



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

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

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL  
INDUSTRIAL

EDISON MARRESE TAYLOR

PROFESOR GUÍA:  
SR. JUAN DOMINGO VELÁSQUEZ SILVA

MIEMBROS DE LA COMISIÓN:  
SR. FELIPE BRAVO MÁRQUEZ  
SR. ALBERTO CABEZAS BULLEMORE

SANTIAGO DE CHILE  
MARZO 2013

# Resumen Ejecutivo

RESUMEN DE LA MEMORIA  
PARA OPTAR AL TITULO DE  
INGENIERO CIVIL INDUSTRIAL  
POR : EDISON MARRESE TAYLOR  
FECHA: 08/03/2013  
PROF. GUIA: SR. JUAN VELÁSQUEZ

El objetivo de este trabajo es diseñar e implementar una aplicación de web opinion mining para encontrar preferencias sobre productos turísticos en la X Región de Los Lagos, Chile.

Esta aplicación se desarrolló bajo el proyecto FONDEF D10I1198, conocido como *WHA-LE (Web Hypermedia Analysis Long-Term Environment)*, que aborda la situación de la industria del turismo en Los Lagos, donde los operadores turísticos caracterizan la demanda y definen la oferta usando estudios de alcance limitado. Estos estudios no son capaces de cubrir un número representativo de participantes porque se aplican a grupos específicos de personas, dejando la demanda potencial proveniente de fuera de la región sin estudiar. Dada esta situación, se torna importante considerar métodos alternativos de estudio.

Con el explosivo crecimiento de la Web 2.0, la cantidad de información disponible online es hoy inmensa. Este trabajo ofrece un enfoque que considera una nueva alternativa para descubrir preferencias de clientes sobre productos turísticos, particularmente hoteles y restaurants, usando opiniones disponibles en la Web en la forma de *reviews*. Esta tarea presenta desafíos importantes, principalmente por el hecho de que los datos son variables en el tiempo y están frecuentemente dispuestos en una forma semi-estructurada.

En este contexto, web opinion mining o WOM ofrece un conjunto de técnicas para analizar datos de opiniones y definir una estructura a partir de ellos. En particular, aspect-based opinion mining propone dividir las opiniones en *aspectos*, tópicos importantes o representativos que, en el caso de los *reviews* de productos, se conciben como componentes o atributos de cada producto con su respectiva orientación sentimental. Este trabajo propone que los reviews en la Web contienen información valiosa sobre productos turísticos y que, mediante la aplicación de algoritmos de aspect-based opinion mining a estos reviews, es posible descubrir las preferencias de los consumidores sobre dichos productos. Esta información, una vez extraída, puede ser usada por diferentes actores en una industria, particularmente, la del turismo en Los Lagos.

El diseño de la aplicación propuesta incluyó modelar las opiniones, generar algoritmos específicos para extraer estas opiniones desde los reviews, crear de corpus linguistico para evaluar el desempeño de los algoritmos y proponer una arquitectura de software para la aplicación en sí. La implementación consistió en desarrollar el software propuesto usando Python.

Los resultados mostraron que los reviews de productos turísticos disponibles en en sitios web contienen información valiosa sobre las preferencias de los consumidores y que estas pueden encontrarse usando un enfoque de aspect-based opinion mining. Sin embargo, en promedio, los algoritmos sólo fueron capaces de extraer un 35% de los *aspectos*, aunque mostraron ser muy efectivos en determinar la orientación sentimental, obteniendo una precision y recall promedio de un 90%.

# Abstract

ABSTRACT OF THE THESIS  
TO OBTAIN THE TITLE OF  
CIVIL INDUSTRIAL ENGINEER  
BY: EDISON MARRESE TAYLOR  
DATE: 08/03/2013  
GUIDE PROF.: MR. JUAN VELÁSQUEZ

The main objective of this study is to design and implement a web opinion mining-based application to find customer preferences about tourism products in the X Región de Los Lagos, Chile.

The application was developed under project FONDEF D10I1198, known as *WHALE (Web Hypermedia Analysis Long-Term Environment)*, which embraces the current situation in the tourism industry in Los Lagos, where tourism operators characterize the tourism demand and define supply configurations using studies with limited scope. These studies fail to cover a significantly representative number of participants because they are applied to specific groups of people, leaving the potential demand from the outside of the region unstudied. Thus, it seems worthwhile to consider alternative possibilities to do so.

With the explosive growth of the Web 2.0, the amount of information that is now available on line is huge. This research offers an approach to a new alternative for discovering consumer preferences about tourism products, particularly hotels and restaurants, using opinions available on the Web in the manner of reviews. The task presents significant challenges, mainly because data is time-varying and frequently disposed in a semi or unstructured way.

In this context, web opinion mining or WOM, offers a set of techniques to deal with data about opinions and perceive a structure within this disorganized data. In particular, aspect-based opinion mining proposes dividing input texts into *aspects*, important or representative topics that, in the case of product reviews, are conceived as components or attributes of the reviewed product with their respective sentiment orientation. This study proposes that tourism reviews on the web contain valuable information about tourism products which, by applying aspect-based opinion mining algorithms to these reviews, it is possible to discover customer preferences about the reviewed products. The information, once extracted, could be used for different stakeholders, specifically the tourism industry in Los Lagos.

The design of the application included modeling opinions and building specific algorithms to extract them from reviews, generating linguistic corpora to evaluate the performance of the algorithms and proposing a software architecture for the application. Implementation consisted of developing the proposed software using Python.

Results showed that tourism product reviews available on web sites contain valuable information about customer preferences that can be extracted using an aspect-based opinion mining approach. However, on average, the algorithms were only capable of extracting 35% of the *aspects*, although they proved to be very effective in determining the sentiment orientation of opinions, achieving a precision and recall of 90%.

*It is known that there are an infinite number of worlds, simply because there is an infinite amount of space for them to be in. However, not every one of them is inhabited. Therefore, there must be a finite number of inhabited worlds. Any finite number divided by infinity is as near to nothing as makes no odds, so the average population of all the planets in the Universe can be said to be zero. From this it follows that the population of the whole Universe is also zero, and that any people you may meet from time to time are merely the products of a deranged imagination.*

*- Douglas Adams, The Restaurant at the End of the Universe*

# Acknowledgements

Quisiera agradecer sinceramente a todos aquellos han recorrido conmigo parte del largo camino que hoy hace posible este trabajo. A todos los que me han brindado una mano amiga cuando hizo falta, a los que han querido confiar en mí, a todos los que han estado conmigo en las buenas y en las malas, desde un principio, y a todos a quienes les debo ser el que soy.

Pero, principalmente, a todos aquellos que me han regalado su sonrisa. Porque, al final del camino, todo lo que queda es lo que hemos hecho por otros y no por nosotros mismos. Al final, solo hace falta sonreír, arrastrados por los recuerdos.

Edison Marrese Taylor

# Index

<b>Resumen Ejecutivo</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Work Context . . . . .	1
1.1.1 Current situation and opportunities in Los Lagos . . . . .	2
1.1.2 Project Description and Justification . . . . .	3
1.2 Objectives . . . . .	4
1.2.1 General Objective . . . . .	4
1.2.2 Specific Objectives . . . . .	4
1.3 Research Hypothesis . . . . .	4
1.4 Methodology . . . . .	5
1.5 Expected Results . . . . .	5
1.6 Contributions . . . . .	6
1.7 Contents Structure . . . . .	6
<b>2 Conceptual Framework</b>	<b>7</b>
2.1 The World Wide Web . . . . .	7
2.1.1 HyperText Markup Language and Tags . . . . .	7
2.1.2 Web Data . . . . .	8
2.2 Data Mining and Web Mining . . . . .	8
2.2.1 Performance Evaluation of Supervised ML Algorithms . . . . .	10
2.2.2 Itemset Mining . . . . .	11
2.2.3 Text Mining and Natural Language Processing . . . . .	12
2.3 Web Opinion Mining . . . . .	18
2.3.1 Data Sources for WOM . . . . .	20
2.3.2 Generic Problems . . . . .	23
2.3.3 Best Known Approaches . . . . .	23
2.3.4 Process Steps . . . . .	24
<b>3 Opinions and preferences</b>	<b>31</b>
3.1 Initial Concepts . . . . .	31
3.2 Models . . . . .	33
3.2.1 Opinion Model . . . . .	34
3.2.2 Opinion Extraction Model . . . . .	36
3.3 Opinion Mining Algorithms . . . . .	39
3.3.1 <i>Aspect expression</i> extraction . . . . .	40
3.3.2 Determination of the Opinion Orientation . . . . .	42
3.4 Finding customer preferences through opinions . . . . .	47
3.4.1 Using Aspect-Based Opinion Summary . . . . .	47
3.4.2 Using Regressions . . . . .	49
<b>4 Design</b>	<b>52</b>

4.1	General Requirements	52
4.2	Annotated Corpus Structure	53
4.2.1	Corpus Description	53
4.2.2	Annotation Structure	54
4.2.3	Tagging Methodology	55
4.3	Performance Evaluation Parameters	59
4.3.1	Aspect Extraction	60
4.3.2	Sentence Subjectivity Classification	61
4.3.3	Aspect Sentiment Classification	61
4.4	Application Design	62
4.4.1	Data Collection Module	63
4.4.2	Results Visualization Module	66
4.4.3	Opinion Mining Module	70
4.4.4	Performance Evaluation Module	71
4.4.5	Data Persistence Module	72
<b>5</b>	<b>Implementation</b>	<b>77</b>
5.1	Development Tools	77
5.2	Data Persistence Module	79
5.2.1	RVM Data Model	79
5.2.2	PEM Data Model	83
5.3	Data Collecting Module	86
5.3.1	Data for Corpora	88
5.4	Opinion Mining Module	88
5.4.1	Aspect Extraction Sub-Module	89
5.4.2	Orientation Finder Sub-Module	90
5.5	Performance Evaluation Module	92
5.6	Results Visualization Module	93
<b>6</b>	<b>Results</b>	<b>96</b>
6.1	Annotated Corpora	96
6.1.1	Annotated Hotels Corpus	97
6.1.2	Annotated Restaurants Corpus	98
6.2	Algorithm Performance Evaluation	100
6.2.1	Performance on the Hotels Corpus	100
6.2.2	Performance on the Restaurants Corpus	101
6.2.3	Average Performance	102
6.3	Application Features	104
<b>7</b>	<b>Conclusions</b>	<b>109</b>
7.1	Future Work	111
	<b>Appendixs</b>	<b>113</b>
A	Paper: <i>“Identifying Customer Preferences about Tourism Products using an Aspect-Based Opinion Mining Approach”</i>	113
	<b>References</b>	<b>124</b>

# Index of Tables

2.1	Classification results vs. real values for a specific class: the confusion matrix.	10
4.1	Details about corpora used by Liu and Hu to evaluate the performance of their algorithms.	54
4.2	Bing Liu’s tagging structure for an evaluation corpus.	55
4.3	Corpus annotation structure proposed in this work.	55
4.4	Tagged sentences example.	59
5.1	Description of table entity for the RVM Data Model.	80
5.2	Description of table comment for the RVM Data Model.	81
5.3	Description of table aspect for the RVM Data Model.	81
5.4	Description of table originalaspect for the RVM Data Model.	81
5.5	Description of table aspect_originalaspect for the RVM Data Model.	82
5.6	Description of table hsentence for the RVM Data Model.	82
5.7	Description of table hsentence_aspect for the RVM Data Model.	82
5.8	Description of table adjective for the RVM Data Model.	82
5.9	Description of table aspect_adjective for the RVM Data Model.	83
5.10	Description of table util for the RVM Data Model.	83
5.11	Description of table entity for the PEM Data Model.	84
5.12	Description of table model for the PEM Data Model.	84
5.13	Description of table sentence for the PEM Data Model.	84
5.14	Description of table aspect for the PEM Data Model.	85
5.15	Description of table hsentence for the PEM Data Model.	85
5.16	Description of table hsentence_aspect for the PEM Data Model.	85
6.1	Hotels Corpus Details.	97
6.2	<i>Aspect expressions</i> from the hotels corpus.	98
6.3	Restaurants Corpus Details.	99
6.4	<i>Aspect expressions</i> from the restaurants corpus.	99
6.5	Performance summary of the best model on the hotels corpus.	101
6.6	Performance summary of the best model on the restaurants corpus.	102
6.7	Average performance obtained in both corpora, compared with Liu’s results.	103



# Index of Figures

2.1	Comparison of tree stemming algorithms. . . . .	14
2.2	Alphabetical list of part-of-speech tags used in the Penn Treebank Project. . . . .	16
2.3	Example of the results of applying a parsing procedure. . . . .	17
2.4	Example of a a set of chunks extracted from a sentence. . . . .	17
2.5	CoNLL-2000 Chunking Notation . . . . .	18
2.6	Poll example in Apple web site. . . . .	21
2.7	Bar chart showing a visual comparison of opinions on two products. . . . .	29
2.8	Opinions towards four persons in March 2000, during the president election in Taiwan. Candidate A was elected. . . . .	30
3.1	Preview of the product attributes identification process developed by Decker and Trusov. . . . .	50
4.1	General design of the Web Opinion Mining Application. . . . .	63
4.2	Data Collection Module use cases. . . . .	66
4.3	Results Visualization Module use cases. . . . .	67
4.4	Aspect-Based summary bar chart example for the entity <i>hotel</i> . Source: Own preparation. . . . .	68
4.5	Screen shot of the TinkTag website, showing its special impression-based chart on Sebastián Piñera (entity). . . . .	68
4.6	Adjective bubble chart examples for the <i>entity</i> hotel and its <i>aspects</i> room and area. . . . .	69
4.7	Tagging interface of Opinion Observer. . . . .	69
4.8	Opinion Mining Module use cases. . . . .	71
4.9	Performance Evaluation Module use cases. . . . .	72
4.10	Data Persistence Module use cases. . . . .	73
4.11	E-R model of the database for the application. . . . .	73
4.12	E-R model of the database for performance evaluation. . . . .	74
5.1	E-R Data Model for the RVM. . . . .	80
5.2	E-R Data Model for the PEM. . . . .	83
5.3	Models class diagram. . . . .	86
5.4	DCM class diagram. . . . .	87
5.5	Complementary class diagrams for the OMM. . . . .	89
5.6	Class diagram of the <i>aspect_extractor</i> package. . . . .	91
5.7	Class diagram of the <i>orientation_finder</i> class. . . . .	92
5.8	PEM class diagram. . . . .	92
5.9	Website structure design. . . . .	95
6.1	Best fitting statistical distributions (continuous and discrete) for the number of sentences in hotels reviews. . . . .	98
6.2	Best fitting statistical distributions (continuous and discrete) for the number of sentences in restaurants reviews. . . . .	99
6.3	Sensitivity analysis on the hotels corpus. . . . .	100

6.4	Sensitivity analysis on the restaurants corpus. . . . .	102
6.5	Home page of the developed web opinion mining application. . . . .	104
6.6	Screen shots of the process of extracting and inserting opinions for an entity in the developed application. . . . .	105
6.7	Screen shot of the page showing aspect-based summaries for hotels (above) and restaurants (below). . . . .	106
6.8	Screen shot of the page showing positive opinions (sentences) for the entity hotel. . . . .	107
6.9	Screen shots of the special pages for <i>aspect expressions</i> , showing adjective bubble charts for lake view for hotels (above) and for pizza for restaurants (below). . . . .	108

# Chapter 1

## Introduction

### 1.1 Work Context

This work has been carried out under project FONDEF D10I1198: “*Desarrollo de una plataforma tecnológica genérica, basada en Web Intelligence, de apoyo al diseño y aplicación de mejores estrategias de creación de valor en la industria de los servicios: experiencia demostrativa en el clúster de turismo de la región de Los Lagos*”.

As the name shows, the project encompasses the current situation in the tourism industry in Los Lagos, where tourism operators (enterprises and government institutions) characterize tourism demand using polls and interviews. On the other hand, the supply configuration is made using demand studies with limited scope. All these instruments do not seem to be a good contribution when it comes to boosting tourism in the area and increasing the flow of visitors.

Taking this into consideration, the project, denominated *WHALE (Web Hypermedia Analysis Long-Term Environment)*, emerges from the need to find a better way to characterize the demand, promote the supply and stimulate tourism activities in the X Región de Los Lagos. Thus, the platform proposed by the *WHALE* project is conceived as a consolidated and accessible data repository that supports stakeholders in the industry by allowing them to make better decisions intended to increase the number of tourists. The platform integrates Web Intelligence techniques for characterizing demand and communicating a higher added-value offer.

Formally, the development of the project considers investigating three problems, namely:

1. Demand characterization.
2. Supply and promotion of tourism services and products
3. More and better quality management indicators for the tourism industry.

As of 2012, the project is being developed in the Industrial Engineering Department of the University of Chile. The team that comprises the project, composed of professionals from varied scientific disciplines, supported all the work presented here.

### 1.1.1 Current situation and opportunities in Los Lagos

Los Lagos is one of the fifteen Regions existing in Chile. It belongs to Chilean Patagonia and according to the 2002 national census, has a surface area of 48,583.6 km<sup>2</sup> and a population of 716,739 inhabitants. In relation to tourism activities, as stated in [1], Los Lagos appeared in the fifth place in the amount of visits during 2010, receiving a total amount of 200,040 persons, a figure that results in a 1.1% increment from 2009.

Faced with such a number of visitors, the benefits of a better characterization of demand and a better promotion of supply would be considerable. The existence of an uncountable number of companies with low technology and technical implementation makes it easy to understand that the emergence of a technological platform such as *WHALE* would definitely make a difference in the development of the market, improving the current situation of the tourism industry.

Also, consider the fact that tourism enterprises in Los Lagos present, in 50% of the cases, incomes lower than 4 million pesos a month in peak or high season. Obviously, this number results even lower during low season, where 60.7% of the enterprises receive an income of less than \$800,000 pesos a month [1]. In other words, since the market is mostly composed of small enterprises that lack efficient mechanisms of estimating the demand or promoting supply, it ends up being very difficult to protect against the variability of visitors during the year by developing better offers to clients according to what customers are really looking for in Los Lagos.

Given this situation, great opportunities to exploit new channels of collecting data appear that help solve these problems. In particular, the implementation of advanced techniques in *Web Intelligence* would allow enterprises to increase their current added value, increase the tourist's level of expenditures in the region and reduce the effects of demand variability.

The most important reason that justifies the use of these techniques in this context relies on the fact that the Web has become a huge base of information regarding various topics affecting many industries, particularly tourism. Indeed, according to [2], the Web is a vast collection of heterogeneous, unclassified, time-varying, semi-structured and high dimensionality data, which allows it to be considered as the largest open and democratic publishing system around the world [3]. Considering the high dimensionality of this huge repository, the task of finding what they are looking for has become harder for users. In that context, *Web Intelligence* proposes a set of diverse techniques that try to obtain not only data and information from the Web, but also knowledge and wisdom [4].

For instance, by applying *Web Mining* techniques on web data, it is possible to discover the user-preferred contents on a website, understand how they reach that piece of information and then propose new website structures that help users find what they are looking for.

Nevertheless, Chile presents little exploitation of web resources, even though an analysis of Web 2.0 generated data would probably permit the capture of real-time tendencies for any market.

### 1.1.2 Project Description and Justification

Nowadays demand characterization and supply promotion are key concepts in any business. If both of them are carried out in the correct manner, any agent inside an industry can achieve important cost reductions and accomplish a better understanding of his customer's needs. Nevertheless, both ideas are anything but simple.

In particular, demand characterization is a complex process by itself. Characterizing demand in any business is always linked to a decrease in the uncertainty about the number of customers that need to be served at any moment in time and about what these customers want in a particular moment. In order to achieve this, what firms need is to understand who their customers are and what they want in the deepest possible sense.

In the case of the tourism industry in Los Lagos, demand characterization is implemented using traditional tools like polls, focus groups and interviews. These approaches present a number of difficulties that decrease the quality of the obtained results, particularly, in three aspects:

- Measuring the satisfaction level of users, customers or clients, because they hide or distort data they deliver (even unintentionally.) As a consequence, this point makes it very difficult to correctly identify what customers really want or need.
- Covering a significantly representative number of participants, because the means are limited or applied to specific groups of people. In general, enterprises today collect perceptions from tourists that are currently visiting Los Lagos, leaving the important potential demand from the outside of the region and country unstudied. This situation occurs although, according to figures from INE, Los Lagos is the second region (after Región Metropolitana) with the biggest amount of international passengers that stay in tourist facilities, receiving more than 143,000 travelers annually [5].
- Quantifying or measuring the amount of expected clients, because there is no platform that allows tourist operators to know the number of past visitors (to estimate the future ones) or a system that permits the potential tourist to notify his visit in advance, for example by simply making a reservation. Only 39% of the companies in Los Lagos have a website and only 48.1% of them have a reservation system, which in most cases simply corresponds to a phone, fax or email reservation [6].

The application of *Web Intelligence* techniques, which by processing textual sources are capable of identifying important or salient topics that would not emerge with traditional methods (like focus groups, for instance) [7], presents a direct solution for the problems mentioned before.

Adapting these techniques to a specific industry, such as tourism in Los Lagos, represents

a challenge both in a research and development context. In relation to these challenges, this study is intended to be a real and valuable contribution.

## 1.2 Objectives

This work pursues the following objectives:

### 1.2.1 General Objective

Design and implement a web opinion mining application to find customer preferences about tourism products in the X Región de Los Lagos

### 1.2.2 Specific Objectives

1. Study and analyze state-of-the-art web opinion mining techniques.
2. Define an opinion model that permits the generation of a structure from the unstructured content on web pages and find customer preferences.
3. Using state-of-the-art techniques as a basis, develop web opinion mining algorithms to extract opinions from the web, based on the proposed models.
4. Design a web opinion mining application that implements all the theory proposed, using classic software engineering tools such as UML and E-R models. The design needs to include a system of obtaining opinionated documents from the web, a logic structure to extract and save opinions and a visual layer that shows the results to the user.
5. Design and create an annotated corpus that permits the evaluation of the performance of aspect-based opinion mining algorithms in the tourism domain, particularly for hotels and restaurants.
6. Implement the web opinion mining application using Python, NLTK and the Django Framework.
7. Evaluate how the web opinion algorithms perform in the tourism domain.

## 1.3 Research Hypothesis

With the explosive growth of the Web 2.0, the amount of information that is now available on the web is huge. As stated before, the application of web mining techniques on specific web data, permits the gaining of insights about how users behave when browsing, or what users prefer on websites (by detecting important or salient topics.) This work proposes that by applying aspect-based opinion mining algorithms to tourism reviews on the web it is possible to discover customer preferences about the products appearing on them.

As a corollary, it follows that tourism reviews on the web contain valuable information about tourism products. This information, once extracted, could be used for different stakeholders in an industry (in this case tourism) to make better decisions.

## 1.4 Methodology

This work begins with an extensive bibliographic revision of the state-of-the-art techniques in web opinion mining and in the use of web content to develop insights in understanding customer preferences. In particular, the special approach called aspect-based opinion mining will be studied in detail.

In the second place, a set of opinion models will be proposed. These models will help understanding how opinions appear on web pages, how they can be extracted from the web content and how the information obtained could be used to discover customer preferences in opinions. Later, a set of algorithms needed to process opinionated documents from the web and extract opinions from them will be developed. These algorithms will use concepts defined in the models and will behave according to what they propose, but will also consider different ideas found during bibliographic review as inspiration.

Then, taking all these ideas as a basis, a web opinion mining Application will be designed. Considering a list of requirements that ensures the application fulfills the objective of this work, the design will use some classic software engineering tools to show how it will be built, such as UML use cases and E-R diagrams. The designing process includes defining the source of opinionated documents that will be used, specifying all the functionalities the application will have and explaining how the performance of the algorithms will be evaluated.

Then, a little research will be done in order to find and select the best set of tools for implementation and learn how the Python Programming Software, the NLTK package and the Django Framework work. Research will include a review of the most important literature and documentation of the mentioned software. Later, the implementation process is carried out.

Finally, an evaluation of the obtained results will be done. This evaluation will consider measuring the performance of the web opinion algorithms, analyzing the features of the developed application and studying how they impact the research hypothesis.

## 1.5 Expected Results

This work proposes the following expected results:

- A conceptual framework that reviews the most important state-of-the-art techniques in web opinion mining.
- A new model for extracting opinions from opinionated documents.
- Two datasets in CSV format, containing tourism reviews of hotels and restaurants, taken from the Lake District on the website TripAdvisor.
- Two annotated corpora (one for hotels and one for restaurants) to evaluate the performance of aspect-based opinion mining algorithms, in txt format.
- A simple rule-based methodology proposal for the tagging process.

- A web opinion mining application that permits users to gain insights about tourism products in the X Región de Los Lagos, using opinions.

## 1.6 Contributions

Thanks to the research carried out in order to write part of the Conceptual Framework Chapter of this work, it was possible to collaborate with professor Juan D. Velásquez, together with Cristián Rodríguez Opazo, in the Chapter named *Web Opinion Mining* of the book *Advanced Techniques in Web Intelligence - 2*, published during 2012 by *Springer*.

It was also possible to collaborate with professors Juan D. Velásquez, Felipe Bravo Márquez and Yutaka Matsuo, in the paper “*Identifying Customer Preferences about Tourism Products using an Aspect-Based Opinion Mining Approach*”, to be accepted in the International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013. The paper is attached in Appendix [A](#) .

## 1.7 Contents Structure

The rest of this document is structured as follows. In the first place, the Conceptual Framework (Chapter 2) gives all the definitions and explains all the concepts that will be used later. These concepts include some basics about the WWW, data mining, web mining, natural language processing and also web opinion mining.

After definitions have been given, Chapter 3 presents details about the models and algorithms that will be used. This chapter deepens in some specific opinion mining techniques mentioned in the Conceptual Framework, using some ideas found in literature, defining some unclear concepts that were ambiguous and also proposing some new approaches. All these techniques and models will be the basis on which to build the proposed application.

Chapter 4 specifies the requirements that the application needs to fulfill and explains how it is designed. The design will be proposed in the most abstract possible manner, including a software architecture, data models and a set of methods to evaluate the performance of the opinion mining algorithms, considering two annotated corpora with reviews about restaurants and hotels that will be used as gold standard.

Finally, Chapter 5 presents all the obtained results, including details about the corpora that were elaborated, a complete performance evaluation of the algorithms and screen shots of the application. Chapter 6 presents conclusions obtained from this work.



# Chapter 2

## Conceptual Framework

### 2.1 The World Wide Web

The World Wide Web, also known as “the WWW,” “the Web” or “W3,” is a huge collection of interconnected documents, available through the Internet. It began as a networked information project at CERN, during the early 90’s, when a body of software, hardware, and a set of various protocols were defined. Through the use of hypertext (as explained in the next section) and multimedia techniques, the web was conceived as easy for anyone to surf, browse and contribute to. Thanks to this simple principle, the WWW has now become a universe of network-accessible information [8].

The basic operation of the World Wide Web relies on three main components:

- A system of globally unique identifiers for resources: the Uniform Resource Locator (URL) and Uniform Resource Identifier (URI)
- A publishing language named HyperText Markup Language or HTML
- A Transfer Protocol designed for HyperText, HTTP.

#### 2.1.1 HyperText Markup Language and Tags

The HyperText Markup Language or HTML is used by web browsers to interpret and compose text, images and other multimedia objects into web pages. Therefore, the HTML code constitutes the main essence of the web pages content, defining the characteristics and properties of every item inside them.

HTML is written in the form of HTML elements, which consist of tags enclosed in angle brackets within the web page content. There are several tags, some of the most important ones are presented in [9]:

- Tags related to the site structure and position of each element or content:
  - div: Mainly used to divide topic blocks.
  - ul,li,ol: Used to show lists of elements.
- Tags related to the text content:
  - h1 to h6: Used to represent titles hierarchies.
  - b, italic: Used to format text.
  - p, table: Used to indicate and separate paragraphs and draw tables.
- Tags related to multimedia objects embedded in the page:
  - img, object: Used to insert images or other objects (like sounds or animations.)
- Tags related to hyperlink generation.
  - a: Used to link a web document to another.

It is important to establish that the purpose of a web browser is to read these tag-based HTML documents and generate web pages. So, even though the browser does not display the HTML code, it uses the tags to interpret the content of the page that is being shown, which is frequently available in the form of source code.

## 2.1.2 Web Data

As shown before, the HTML tags generate the structure and save the content of each web page. Nevertheless, the content one could find on web sites is not the only kind provided by the Web.

As stated in [3], from all possible web data, three types have a special significance, because they could provide the chance to discover valuable information: web logs, web pages and the web hyperlinks structure. However, most of this data lacks a defined structure, is often unlabeled and includes a lot of noise. So, in order to obtain this valuable information from them, they must be pre-processed.

## 2.2 Data Mining and Web Mining

According to Gronescu [10], Data Mining, also known as Knowledge Discovery in Databases or KDD, has three generic roots: Statistics, Artificial Intelligence (AI), including Machine Learning, and Database systems (DBS.) But, even though these key roots are clearly defined, giving a unique definition of KDD is quite hard. Authors have proposed many approaches so far, nevertheless, the most common definition was probably the one given in [11], declaring that KDD is the process of non-trivial extraction of information from data, information that is implicitly present in that data, previously unknown and potentially useful for the user. This valuable information present in data is found in the manner of patterns, which are useful when applied to a determined problem or business context.

One of the most important roots of Data Mining is Machine Learning or ML, which is a branch of Artificial Intelligence. More specifically, ML is a discipline concerned with the development of algorithms that take as input empirical data to carry out two tasks:

1. Identify (i.e., quantify) complex relationships thought to be features of the underlying mechanism that generated the data.
2. Employ these identified patterns to make predictions based on new data.

In this context, data can be seen as instances of the possible relations between observed variables, so the algorithm acts as a machine learner which studies a portion of the observed data (the training data) to capture characteristics of interest of the data's unknown underlying probability distribution, and employs the knowledge it has learned to make intelligent decisions based on new input data [12].

Machine Learning algorithms can be classified in several categories, depending on the desired results. Here, it is interesting to mention two of them:

- **Supervised Learning:** These are algorithms that generate a mathematical function to map inputs to a set of desired outputs. The desired outputs are frequently called labels, because they are often a result of a human tagging process.
- **Unsupervised Learning:** These algorithms do not attempt to map an input into a known set of outcomes, but simply model a set of inputs.

*Web Mining* techniques emerge as a result of the application of the data mining theory to discover patterns in web data. This task is not trivial because web data exists in a heterogeneous, unclassified, time-variable and semi-structured manner, with high dimensionality.

Taking into consideration the main types of web data, three different general *Web Mining* techniques can be distinguished [13]:

- **Web Content Mining (WCM):** aims to mine the web content or, in other words, the information the user of a web page sees on the screen. Among this data, text plays an important role, but it is also possible to find images, sounds, videos and other objects.
- **Web Structure Mining (WSM):** aims to mine the organization of pages on a web site. Web structure data mostly includes HTML tags that generate *hyperlinks* between web pages. With this information, a graph can be constructed and studied in detail.
- **Web Usage Mining (WUM):** is the mining process that tries to discover user browsing preferences on a web site. In this context, web usage data is represented by web server logs, which store every operation they carry out. In addition to this, user data about registered users on some web pages permits making a more complete analysis.

## 2.2.1 Performance Evaluation of Supervised ML Algorithms

For any classification problem, there is a set of measures that evaluates the degree of closeness of predicted classes to the actual (true) values. Then, for each class that the classification problem has, the evaluation measures are obtained comparing the classification results with the real values. In particular, the terms *true positives*, *true negatives*, *false positives*, and *false negatives* compare the results of the classifier under test with trusted external judgements. The terms as defined for a specific class are shown in table 2.1.

	<b>Actual Positives</b>	<b>Actual Negatives</b>
<b>Classified as Positives</b>	True Positives (TP)	False Positives (FP)
<b>Classified as Negatives</b>	False Negatives (FN)	True Negatives (TN)

Table 2.1: Classification results vs. real values for a specific class: the confusion matrix.  
Source: Own elaboration.

In the table, frequently called the confusion matrix or contingency table, the terms positive and negative refer to the classifier's prediction expectation, and the terms true and false refer to whether that prediction corresponds to the external judgement. Considering these definitions, four measures are frequently considered:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

$$F - measure = 2 \frac{TP \times FN}{TP + FN} \quad (2.4)$$

Then, in a classification problem, precision for a class is the number of items correctly labeled as belonging to the positive class divided by the total number of elements labeled as belonging to the positive class. On the other hand, recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class.

From the given definitions, it follows that precision does not say anything about the number of items that were not labeled correctly, while recall does not say anything about how many other items were also incorrectly labeled as belonging to the positive class. For that reason, precision and recall are not usually discussed in isolation. Instead, both are combined into a single measure, such as the F-measure, the weighted harmonic mean between them.

Besides classification tasks, these measures are also used in the context of Information Retrieval, a scientific research field that tries to develop ways of obtaining information resources relevant to an information need, from a collection of information resources. In this case, a high recall means that an algorithm returned most of the relevant results, i.e. a high percentage of the total number of items that were needed to be found were retrieved. High precision means that an algorithm returned more relevant results than irrelevant, i.e. the algorithm extracted made few irrelevant or undesirable results.

Outside of Information Retrieval, for instance in classification tasks, the application of recall, precision and f-measure are argued to be flawed as they ignore the true negative cell of the contingency table (it does not appear on any of their equations.) This problem is usually solved by including accuracy when analyzing results. Nevertheless, accuracy does not deliver good insights alone either and should be used together with precision, recall and f-measure. Accuracy could be a very tricky and misleading measure in some cases, since it does not consider the performance of the negative class generating what is known as the *accuracy paradox*. This paradox states that predictive models with a given level of accuracy may have greater predictive power than models with higher accuracy.

When the amount of true positives is smaller than the amount of false positives for a particular test, the accuracy increases when this test is discarded and replaced by a test where the test outcome for all elements is negative. The same holds when the amount of true negatives is smaller than the number of false negatives and the test outcome for all elements becomes positive [14]. From this, it follows that it may be better to avoid the accuracy metric in favor of other metrics, such as precision and recall. In situations where the minority class is more important, f-measure may be more appropriate.

## 2.2.2 Itemset Mining

Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. As stated by Goethals in [15], the problem is formally defined as follows:

Let  $\mathcal{I}$  be a set of items. A set  $X = \{i_1, \dots, i_k\} \subseteq \mathcal{I}$  is called an *itemset* if it contains  $k$  items.

A transaction over  $\mathcal{I}$  is a couple  $T = (tid, I)$  where  $tid$  is the transaction identifier and  $I$  is an *itemset*. A transaction  $T = (tid, I)$  is said to support an *itemset*  $X \subseteq \mathcal{I}$ , if  $X \subseteq I$ .

A transaction database  $D$  over  $\mathcal{I}$  is a set of transactions over  $\mathcal{I}$ . Whenever it is clear from the context  $\mathcal{I}$  will be omitted. The cover of an itemset  $X$  in  $D$  consists of the set of transaction identifiers of transactions in  $D$  that support  $X$ :

$$cover(X, D) := \{tid | (tid, I) \in D, X \subseteq I\} \quad (2.5)$$

The support of an itemset  $X$  in  $D$  is the number of transactions in the cover of  $X$  in  $D$ :

$$\text{support}(X, D) := |\text{cover}(X, D)| \quad (2.6)$$

The frequency of an itemset is the probability of  $X$  occurring in a transaction  $T \in D$ :

$$\text{frequency}(X, D) := P(X) = \frac{\text{support}(X, D)}{|D|} \quad (2.7)$$

An itemset is called frequent if its support is no less than a given absolute minimal support threshold  $\sigma$ ,  $0 \leq \sigma \leq |D|$ . Then, given a set of items  $I$ , a transaction database  $D$  over  $I$  and minimal support threshold  $\sigma$ , the problem of itemset mining is to find  $F(D, \sigma)$ , the collection of frequent itemsets in  $D$ .

$$F(D, \sigma) := \{X \subseteq I \mid \text{support}(X, D) \geq \sigma\} \quad (2.8)$$

Together with the introduction of the frequent set mining problem, also the first algorithm to solve it was proposed. Shortly after that, that algorithm was improved and called Apriori in 1995 in [16]. The algorithm actually does not only solve the frequent itemset mining problem, but also the complete association rule mining problem (not interesting for this work) [15]. Another algorithm, also commonly used in applications is the Eclat algorithm [17], proposed by Zaki around 1997.

### 2.2.3 Text Mining and Natural Language Processing

Text Mining is the field of Data Mining that deals with text data. In fact, it is sometimes alternately referred to as text data mining. Usually, text mining refers to the process of deriving high-quality information from text, usually involving the process of structuring the input text, deriving patterns within the structured data and finally evaluating the output. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization and learning relations between named entities.

Besides analytical methods, the application of Natural Language Processing (NLP) is one of the most useful tools in text mining. As stated in [18], many definitions of Natural Language Processing exist and probably there is not a single agreed-upon definition that would satisfy everyone. Nevertheless, as it is important to establish common ground, it is generally declared that the goal of NLP is to accomplish human-like language processing. As such, NLP is related to the area of human-computer interaction.

Thus, Natural Language Processing is a field of computer science, artificial intelligence and also linguistics, that is concerned with the interactions between computers and human (na-

tural) languages, i.e., language as spoken by humans. NLP is based on a wide set of theories and technologies, and is nowadays a very active area of scientific research and development.

## Tokenization

Tokenization is the process of breaking a chain of text up into smaller pieces or components. These components, or *tokens*, could represent meaningful text elements like words, phrases and similar, or could be a different meaningless text unit.

In general, tokenization is applied in order to carry out a more complex process involving text, where the list of tokens obtained from the input is used for further processing, such as NLP or text mining; tokenization is useful in a wide range of domains, obviously including linguistics and computer science.

Despite tokenization being useful in so many different fields of study, the process it implies is always quite the same, consisting of taking raw text as input, then applying a set of segmentation rules and finally generating the set of output tokens. Naturally segmentation rules (and also the tokens) will depend on the domain and the particular needs.

In NLP, many approaches have been developed to achieve tokenization. Probably two of the most relevant techniques are based in Machine Learning and in the use of regular expressions, which provide a concise and flexible way to recognize strings of text.

One special issue in this topic is sentence tokenization, or sentence boundary disambiguation. In English, the use of punctuation provides a reasonable approximation for sentence breaking, but the problem is not trivial due to the use of abbreviations and other special characters that could also terminate a sentence. In this context, [19] provides an unsupervised learning method, called *Punkt Sentence Tokenizer*, which seems to have achieved the best performance in English so far.

## Stopwords Deletion

Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user query need to be eliminated. For that reason, these words, called stopwords, are frequently excluded from the extracted vocabulary in different ways [20]. In general, stopwords are filtered out prior to or after processing of natural language data in the form of text.

There are various strategies to determine a list of stopwords or stop list. There is no such thing as a definitive stop list, but rather, lists are compiled by choosing the most common terms appearing in the documents that are being processed, also according to the semantic content relative to the domain of the documents. In addition, some domain-independent lists have been developed and are publicly available, for instance, the MySQL full-text stopwords, available in the software and on <http://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html>. Despite that, lists are usually composed of meaningless words like articles, pronouns and prepositions.

Using a stop list significantly reduces the number of postings that a system has to store, reducing data dimensionality. Most of the time, not indexing stopwords does little harm and generates an improvement in the efficiency of some information retrieval tasks.

## Stemming and Lemmatization

For grammatical reasons, words present multiple forms in documents, such as “*organize*”, “*organizes*”, and “*organizing*”. Additionally, there are families of derivationally related words with similar meanings. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. Considering this, the goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base or root form [20]. Nevertheless, these two concepts differ in the flavor.

In the first place, stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time. The stem, or resulting term, does not need be identical to the morphological root of the word, being usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since 1968.

Nowadays, probably the most common algorithm for stemming English, and one that has repeatedly been shown to be empirically very effective, is Porter’s algorithm, developed by Martin Porter 1980. The algorithm that Porter proposed basically consists of 5 phases of word reductions that are applied sequentially. Each phase has certain rules that are automatically selected and applied to the text. Porter also developed the Snowball Programming Language [21], a small string-handling language that facilitates working with texts. Snowball is a language in which stemming algorithms and other similar processes can be easily represented since string patterns are used to control the flow of the program. More details about the Porter Stemmer and Snowball can be found on his official site <http://snowball.tartarus.org/>. Other stemmers also exist, including the Lovins stemmer (1968) and the Paice/Husk stemmer (1990.) Figure 2.1 presents an informal comparison of the different behaviors of these stemmers.

*Sample text:* Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

*Lovins stemmer:* such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpre

*Porter stemmer:* such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

*Paice stemmer:* such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Figure 2.1: Comparison of tree stemming algorithms.

Source: [20].

On the other hand, lemmatization usually refers to a process which does full morphological



analysis to accurately identify the root for each word. Lemmatizers normally aim to remove only inflectional endings and to return to the base or dictionary form of a word, which is known as the lemma.

While stemmers use language-specific rules, but require less knowledge, lemmatizers need a complete vocabulary and morphological analysis to correctly lemmatize words, but return existing and correctly written words. For instance, consider the token *saw*. The Porter stemmer would return just the stem *s*. In this case, although the exact stemmed form does not matter and only the equivalence classes it forms does, the resulting word has lost all meaning. On the other hand a lemmatizer would try to return either *see* or *saw*, depending on whether the use of the token was as a verb or a noun. This certainly seems more intuitive and useful.

## Pos Tagging

Part-of-speech tagging, or POS tagging, is the process that aims to mark up each word in a corpus as corresponding to a particular part of speech, based on both its definition and its relationship with adjacent or related words in a sentence or paragraph.

From the beginning, POS tagging has been directly related with the elaboration of Linguistic Corpora. The first tagging process was performed manually during the 60's using the Brown Corpus [22], one of the the biggest corpora of English for computer analysis. The tagging task, which lasted for several years, finished with the development of a program that automatized the process. This program was continually improved during the following years and by the late 70's, the algorithm was nearly perfect.

Both supervised and unsupervised methods have been proposed, but the first ones are most widely used. In relation to them, there are two main approaches:

- Stochastic Methods: Taking the work with the Brown Corpus as a basis, many statistical approaches have been developed. These techniques include, for instance, the use of Hidden Markov Models (or HMMs), which involve counting cases and making a table of the probabilities of certain sequences, and dynamic programming algorithms, which try to solve the same problem in less time.
- Rule-based Methods: Basically, a technique proposed by Eric Brill in his Ph.D. thesis in 1993 [23]. This technique learns a set of patterns and then applies those patterns rather than optimizing a statistical quantity.

In POS tagging, a special issue is determining the *tag set*, in other words, the annotation system that will be used to mark each possible part of speech. POS tagging work has been done in a variety of languages, and the set of POS tags used varies greatly with each one. Whether a very small set of very broad tags or a much larger set of more precise ones is preferable, the number of tags will depend on the purpose at hand. In the case of automatic tagging, it is obviously better to have smaller tag sets.

Anyway, there are probably only two tag sets that are the most widely used. In the case of American English, the Penn tag set, developed in the Penn Treebank [24] project at the University of Pennsylvania is probably the most common choice. It is also frequently preferred by automatic tagging systems, since it is largely similar to the Brown Corpus tag set, but much smaller. On the other hand, in Europe, tag sets from the EAGLES (Expert Advisory Group on Language Engineering Standards) Guidelines see wide use and include versions for multiple languages. In this work, the Penn tag set will be adopted since it is used by some of the tools that have been chosen. Figure 2.2 shows the list of all tags available.

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP\$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP\$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Figure 2.2: Alphabetical list of part-of-speech tags used in the Penn Treebank Project.

Source: [25].

## Chunking

Finally, syntactic chunking (or simply chunking) deserves a special mention. Chunks correspond in some way, as defined by Abney in 1994 [26], to prosodic patterns. Abney gives

some intuition to define a chunk, indicating that it appears, for instance, that the strongest stresses in a sentence fall one to a chunk, and pauses are most likely to fall between chunks.

Following the intuition of Abney, chunks are defined strictly syntactically and can be understood as textual units of adjacent word tokens which can be mutually linked through unambiguously identified dependency chains with no recourse to idiosyncratic lexical information [27]. Taking Abney’s work as a basis, Gardent states that chunks present a set of properties:

- Chunks are non-overlapping regions of text.
- (Usually) each chunk contains a head, with the possible addition of some preceding function words and modifiers
- Chunks are non-recursive, a chunk cannot contain another chunk of the same category.
- Chunks are non-exhaustive, some words in a sentence may not be grouped into a chunk.
- Noun groups and verb groups are chunks.

Then, chunking consists of dividing a text in syntactically correlated parts of words. From this it follows that chunking is an intermediate step towards full parsing. A parser is capable of assigning a syntactic structure (i.e. discovering the structural relationships between words and phrases) to a string on the basis of a grammar, used to describe the syntax of a language. The following figure shows an example of parsing.

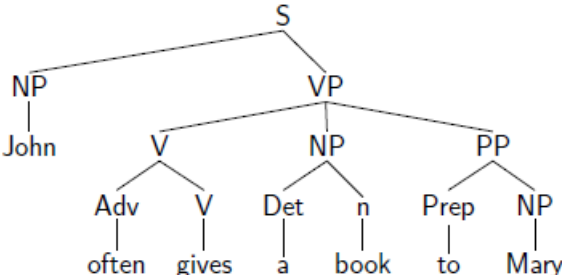


Figure 2.3: Example of the results of applying a parsing procedure.  
Source: Own elaboration.

In contrast to parsing, chunking yields flatter structures than full parsing, generally using fixed tree depth (max depth of 2 vs. arbitrarily deep trees) and does not try to deal with all of language nor attempt to resolve all semantically significant decisions. Figure 2.4 shows an example of what a chunking process aims to extract from a sentence.

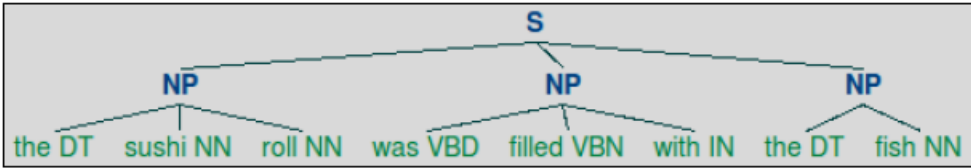


Figure 2.4: Example of a set of chunks extracted from a sentence.  
Source: [28].

Also, as with POS tags for words, categories can be identified when chunking. Most common categories include Noun Phrases, Verb Phrases, Prepositional Phrases, Adjectival Phrases and Adverbial chunks [29]. In addition, different notations have been developed so far. In particular, the notation used in the Conference on Computational Natural Language Learning in 2000 (or CoNLL-2000) will be adopted. An example of this notation is attached below in figure 2.5.

He	PRP	B-NP
reckons	VBZ	B-VP
the	DT	B-NP
current	JJ	I-NP
account	NN	I-NP
deficit	NN	I-NP
will	MD	B-VP
narrow	VB	I-VP
to	TO	B-PP
only	RB	B-NP
#	#	I-NP
1.8	CD	I-NP
billion	CD	I-NP
in	IN	B-PP
September	NNP	B-NP
.	.	O

Figure 2.5: CoNLL-2000 Chunking Notation  
Own elaboration, compiled from [29].

In the figure, the first column contains the current word, the second its part-of-speech tag (frequently needed to find chunks) and the third its chunk tag. The chunk tags contain the name of the chunk type and the special mark B-CHUNK is used for the first word of the chunk, while I-CHUNK is used for each other word in the chunk.

## 2.3 Web Opinion Mining

On many occasions making a good decision requires the opinion of a third person, whether because of insecurity, needing a backup or not having sufficient knowledge of the subject. One then begins to consult for information, details, comparisons and opinions in order to have a better idea on the proposal or concept at hand.

For example, wanting to buy a bike is often consulted on with other people who are more related to or have more experience on the subject, for example regarding which brand is best, what characteristics to be considered, which is more convenient (speed or mountain) and if it is better with or without shocks. After considering all the opinions given in this regard, we eventually make a decision on which bike to buy.

If the foregoing advice is considered in a business plan, it shows that for a customer to be sure about what he is going to consume, either products, services, etc., and avoid spending money needlessly or in error, it is essential to consult someone who has experience in the

area. The result is the concrete idea that opinions are one of the most important indicators of personal decisions when purchasing a product, taking a tour, selecting a hotel to stay in, where to eat, etc. Many people ask their friends or family to recommend products based on their previous experiences, but there are actually more ways to communicate between persons. Thanks to the spread of the Internet and the continued growth of social networks like Twitter, Facebook, and other sites such as blogs or product review pages, it is now possible for users to take opinions and experiences from a bigger circle of people than just family or friends.

In fact, with the inception of the Web 2.0 and the explosive growth of social networks, enterprises and persons are increasingly using the content in these media to make better decisions [30], [31], [32]. More people check the opinions of other shoppers before buying a product, when trying to make a good choice. On the other hand, for organizations, the vast amount of information available in a public manner on the Web could make polls, focus groups and some similar techniques an unnecessary requirement in market research. Results provided by WOM techniques could then represent a real alternative in this context, mostly for traditional questionnaire-based market research methods. Also, not only enterprises would be benefited, but also clients (future customers) that are looking for reviews or opinions about a product or service on the web [33].

Indeed, based on a survey of more than 2,000 U.S. Internet users [34], more than 75% of product review users reported that the review had had a significant influence on their purchase. Consumers reported a willingness to pay from 20% to 99% more for a 5-star-rated item than a 4-star-rated item. In another survey of 475 U.S. consumers [35], over 60% utilized on-line opinions when making purchase decision. More than 59% of consumers used the web to read on-line reviews, ratings of products or brands and research products and features, when buying products which cost between less than \$100 and more than \$1000.

The interest in users feedback about a product or service and the influence it has on them is very important for companies that develop products and services since they can control how their products and their competitors' products were received by the market. As a result, they can determine what things are important to users, what features should improve, and how to modify their advertising strategies in order to attract more users.

On the other hand, views on political decisions or choices are also interesting for politicians since they allow them to evaluate how things are going, what the most important problems to be solved for the people are, whether they are likely to be elected, and so on.

For these reasons, it is interesting to create a tool that can extract a set of opinions and determine what people think about certain products, services, features based on the amount of positive or negative views people have on any of these topics. Depending only on the target object that has been evaluated, the term opinion mining appears in a paper by Dave et al. [36] where the ideal opinion mining tool should be to *“process a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good).”*

WOM is a new tool and has a long road ahead. Thus, giving a unique definition for

WOM is not a simple task because the process final objective is still unclear. It is possible to find many ways to view this problem in literature. However, in any case, the key to extract knowledge from the web, particularly opinions is to find a way of defining a structure from the contents that are disposed in a unstructured manner. Possibilities of defining a structure are unimaginable and each approach will depend on the specific problem that is being attacked. Opinion mining is not an exception to this, however, all existing techniques have some characteristics in common: a set of opinion sources and a list of generic and challenging problems. In the following sections, these two topics are discussed.

### **2.3.1 Data Sources for WOM**

Having already defined the problem of web opinion mining, the natural next question one asks is related to the possible applications of this technique and subsequently, the set of possible web data sources to use as input.

As the Web continues to grow, the number of possible sources for web opinion mining grows rapidly too. Particularly, in the context of WOM, the explosive development of social media plays an important role. Within social media, it is possible to find a variety of different platforms whose content is being increasingly used by individuals and organizations for their decision making [37].

Social media includes web pages such as reviews, forum discussions, blogs, and social networks, like Facebook, Twitter, Foursquare and so on. This variety of sources suggests a heterogeneous mix of structures to work with. As a consequence of this, one needs to specify a different strategy for each source, each one oriented to the particular problem of extracting data, then processing this data and finally discovering valuable information locally within the selected source. Thus, given the mosaic of different structures and procedures for each source, the problem of designing a global methodology is complex and hasn't been widely discussed in literature. The more complete approach to this topic is the idea presented by Bing Liu, from Illinois University, in some publications and with more detail in the book *Web Data Mining*.

The rest of this section is a review of the WOM problem in some particular sources. In first place, the problem of extracting opinions from blogs, forums and news is studied. Then, Twitter is analyzed in order to characterize its content and determine how it can alter the Web Mining strategy to apply to extract knowledge from opinionated documents. Finally, a general and brief characterization of other sources is presented.

#### **Blogs, News and Forums**

Probably, the sources that contain the richest opinionated documents are blogs, news and forums. These sources present some common features that make them a good choice when deciding where to look for opinions, but before any analysis, it is worth mentioning that there are some differences between these sources. News and blogs are two important sources of opinions. The writing of the former is formal in comparison to that of the latter because blog articles expressing personal opinions are often written in casual style. Because of this,

generally speaking, news documents are more objective, while blog articles are usually more subjective. Other important difference refer to opinion holders, which probably belong to different social classes. Opinions extracted from news are mostly from famous people, while opinions expressed in blogs may come from a no name. This issue turns out to be even more important in forums, where it is often difficult to determine the real opinion holder's name [38]. However, when analysing a specific public issue, listing opinions from different sources in parallel can provide many views of the issue, which helps to understand how different social actors react toward the same situation.

Consumer reviews of products are also very similar to blogs, including ungrammatical sentences, incomplete sentences (sentence fragments), short phrases, and missing punctuations. In [39] Liu and Hu state that there are three main review formats on the Web.

- Format (1) Pros and Cons: The reviewer is asked to describe Pros and Cons separately. C—net.com uses this format.
- Format (2) Pros, Cons and detailed review: The reviewer is asked to describe Pros and Cons separately and also write a detailed review. epinions.com and MSN uses this format.
- Format (3) Free: The reviewer writes freely, i.e., no separation of Pros and Cons. Amazon.com uses this format.

In addition, a special format that has been appearing lately, permits user to select the what kind of comment he is adding. Next figure shows an example, taken from a poll in Apple web site.

¿Te gustaría hacerle algún otro comentario a Apple sobre tu reciente experiencia con el servicio del operador de telefonía? (Nota: Límite de 2.000 caracteres)

¿El comentario anterior fue un elogio, una sugerencia o una queja?

Elogio  
  Sugerencia  
  Queja  
  Sin respuesta

Figure 2.6: Poll example in Apple web site.

On the other hand, the most important common feature between news and blogs refers to the document extension, which compared with forums and other sources is way longer. At first glance, this could be seen as an advantage, but long document extensions present a new main problem in opinion mining, i.e., how to determine where an opinion sentence is? To solve this problem, major topic detection techniques are proposed in [38], [40], and some other related publications, to capture main concepts embedded implicitly in a relevant document set.

Finally, in relation to forums, document extensions can be quite different among singular topics or communities, so it is difficult to establish a main tendency within this field. However, a useful and positive feature that appears in a high number of review forums is the post rating system. Post ratings can be used as a predictor of the content semantic orientation or to contrast and validate text processing and analysis results.

## **Twitter**

Twitter is often considered a microblogging platform, but it is also frequently included as a social network. Given the features of this platform, probably both of these considerations have a certain degree of truth, but for the problem of web opinion mining, the fact that Twitter is a microblog is highly relevant. Microblogging is a growing popular communication channel on the Internet, where its users can write short text entrances in a public or private way (to a group of contacts). Messages are extremely short, allowing users to write a maximum of 140 characters on each post, called a tweet. These tweets can be written through the Twitter web interface or through a variety of mobile devices, like smartphones, some cell phones and other devices. These short messages can be seen just as a newspaper headline and subtitle, which makes them easy to produce (write) and process (read). This feature makes microblogs unique when compared to other similar web communication media, like blogs and web pages [41].

As a microblogging social network, the first relevant feature of Twitter's messages are their brevity, which make users look for various special ways to add content in the messages. The most-used approaches are adding content indirectly or trying to use fewer characters to express the same ideas. One important fact on the first topic is the inclusion of links to other web sites to indirectly complement the content of the tweet. In addition to this, it is possible to find a high density of messages containing short URL services, which in fact were created to help Twitter users to include these links with a low number of characters. Another issue refers to the use of Twitter's special characters, like hashtag (#) to denote a particular topic, to call a user, and RT, short for retweet. On the other hand, a different problem associated with tweets is that they are written in colloquial or informal language. In addition to the shortness of messages, this supposes the use of many colloquial symbols or expressions that, in order to be understood as regular text, require preprocessing.

Also, it is interesting to see Twitter as a network of related users, who share opinions among themselves and influence each other. Twitter provides some useful tools that can be used to analyse how a specific topic or opinion tendency is spread through the network, and discover users who influence others or are more easily influenced by another. Based on this, one could be able to, for instance, define clusters and find non-trivial segmentations based on the characterization.

## **Other Media**

There are a lot of other possible sources for WOM. In the context of social networks, Facebook is often proposed as a powerful and complete repository. Nevertheless, the main problem in this case is related to privacy policies and access limitations to the contents. Thus, opinionated documents are not always publicly available, and it is difficult to reach



them.

## 2.3.2 Generic Problems

Since opinion mining implies dealing with texts taken from sources detailed in last section, these texts will not necessarily be written correctly. This issue presents a new set of challenges that are needed to be considered when doing opinion mining. Some these problems are explained below.

1. One first special aspect is that opinions might include a variety of emoticons. This situation often occurs in Twitter. These emoticons help users to express their ideas, feelings or moods in fewer characters, but have a deep impact on text processing. As emoticons aren't considered words, they don't have any structure that helps in extracting its lexical meaning, and are not formally included in any dictionary or language that helps to understand their meaning. Nevertheless, as proposed in [42], emoticons can be successfully used to previously determine a tweet's sentiment orientation, opening a wide new field of investigation. Based on this proposal, in [43], emoticons are used to create a corpus collection strategy, defining two kinds: happy (including "=", ":", ".D" and other similar) or sad (":-(", ":((", "=(", etc.).
2. In second place, spelling mistakes are also common. This implies the implementation of a preprocessing task in order to fix possible mistakes and ensure a correct interpretation of the content. In relation to this, [44] proposes a technique based on the Levenshtein algorithm, which determines a notion of distance between the misspelled and actually meant word. The algorithm can be used combined with a dictionary to rank and then select the most probable letter replacements from a list of previously-generated possible word candidates.
3. Another problem the repetition of one letter in a word. This is mostly done when users want to add any emotive intensification to the text, for instance, repeating vowel letters as in "I loooove you". In this field, [45], proposes a control system based on regular expressions for Spanish with back reference, replacing the appearance of two or more characters by only one letter, excepting the groups "cc", "ll" and "rr", which are commonly used in this language.
4. A last feature is related to the use of Internet language, or "Netspeak", on the messages. In this context, the use of capital letters can be problematic when tokenizing or lemmatizing text during preprocessing.

## 2.3.3 Best Known Approaches

### *A. Aspect-Based Opinion Mining*

Aspect-based opinion mining divides input texts into aspects, also called features or sub-topics in literature. These aspects are usually arbitrary topics that are considered important or representative of the text that is being analyzed. The aspect-based approach is very popular and many authors have developed their own perspectives and models. In relation to this,

Kim et al. gives a good review of historical and state-of-the-art aspect-based developments in [46]. The authors also indicate that the process is commonly made up of three distinct steps:

1. Aspect-Feature Identification, to find important topics in the text (which will then be used to summarize.)
2. Sentiment Prediction, to determine the sentiment orientation on each aspect.
3. Summary Generation, to present processed results in a simple manner.

Although these steps are fairly accurate and wide enough to include most of the existing aspect-based techniques, it is also possible to find some approaches that somewhat integrate them in one single model. Details on this will be given in the next section.

### ***B. Non-Aspect-Based Opinion Mining***

This category includes all the other kinds of opinion mining which do not divide the text into subtopics. In general, non-feature-based techniques simply consider the text as a big object or increase granularity analyzing each paragraph, sentence or phrase. Then, the steps and final result depend on the level chosen. Nevertheless, in general it is possible so consider a generic three-phase process, which is introduced by [40].

- The first phase is Corpora Acquisition Learning, whose aim is to automatically extract documents containing positive and negative opinions from the Web for a specific domain. They propose collecting the corpus by running queries in a search engine, by/or/and entering queries specifying the application domain.
- The second phase is Adjective Extraction. In relation to this task, they propose an automatic extraction of sets of relevant positive and negative adjectives, assuming that adjectives are representative words for specifying opinions. Other equivalent techniques are also proposed in literature.
- The final phase is Classification of new documents. In this case, this is done using the sets of adjectives obtained in the previous phase, calculating the document positive or negative orientation by computing the difference between the number of positive and negative adjectives encountered at document level. Nevertheless, this step can be generalized to any other further level and classification can be obtained using other techniques. They will be reviewed in the next section.

#### **2.3.4 Process Steps**

In this section, a complete list of all the possible steps of all existing opinion mining approaches is presented. Although some of the steps will only be considered under a specific approach, building a list that includes all of them makes sense since the steps follow a natural sequence anyway. For each step, a little description is provided, also mentioning main related publications and authors who have contributed.

## Aspect Identification and Extraction

For aspect-based approaches, the first operation that needs to be performed is the identification of important or representative topics in the text. As stated by Bing Liu in [37], this concept comes from the idea that, in general, opinions can be expressed about anything: a product, service, organization, etc. Then, targets of each opinion, i.e., each object that is being evaluated in each segment of the text could be treated independently. The set evaluated objects, hereinafter the set of aspects underpinning the text, could or could not be known previously, which implies that different problems need to be solved. Also, different people could refer to the same aspect with different words, which brings additional problems to the task. As showed in [46], several techniques have been developed to carry out this step:

- **Pure NLP Techniques:** Some approaches attempt to identify features in the opinion text with the help of NLP-based techniques. Part-of-speech (POS) tagging and syntax tree parsing (chunking) are very common starting points for aspect discovery. In most of the cases, annotated opinion texts are then analyzed using classic data mining techniques. Examples of this are the works of Lu [47], Popescu and Etzioni [48] and Hu and Liu [49].
- **Mining and Statistical Techniques:** As indicated below, classic data mining approach on finding aspects are also used, usually as an attempt to compensate the weaknesses of a pure NLP-based technique. For example, in Hu and Liu’s work, frequent itemset mining methods are applied to accurately determine if a particular word or phrase is a feature or not. In general, as stated by [46], this approach shows reasonable performance, especially with product reviews. Also Archak et al. [50], Popescu and Etzioni 2005 [48] and Decker and Trusov [51] have developed these kinds of techniques.
- **Ontology-Supported Techniques:** Some authors look for aspects by exploiting ontologies. In linguistics and computer science, an ontology represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts. Then by using an ontology, the set of possible aspects is given, and the problem of extracting them transforms into matching. As declared in [52], overall, extracted features correspond exclusively to terms contained in the ontology. In some cases extraction is guided by a manually-built domain ontology [53] but some authors also build the ontology semi-automatically. On the other hand, some other authors simply take known ontologies and use them to find relevant aspects [54], [52].
- **Integrated Approaches:** Finally, there are some integrated approaches that integrate the step of aspect extraction into a complete model, also considering some of the steps explained below.

## Subjectivity Classification

Subjectivity Classification aims at different sub-segments of the text, trying to differentiate sub-segments that include any opinion or evaluation from the ones that do not. The process can be applied to documents, paragraphs or sentences. In [46], Kim et al. indicate that subjectivity classification is different from sentiment classification (see next section) in that the former only aims at finding if an opinion is present or not and does not attempt to

identify the orientation of these opinions.

As mentioned by Liu in [37], existing literature regarding this problem is vast. Hatzivassiloglou and Wiebe [55] did some of the first work in this topic in 2000, attempting to find high quality adjective features by word clustering. In 2003, Riloff and Wiebe [56] used subjective nouns learned automatically from un-annotated data, also Yu and Hatzivassiloglou [57] presented a Bayesian approach to identify if a document is subjective or not. Some other studies consider the same authors and can be found in [58], [59], [60] and [61].

Finally, Kim et al. also indicate that the importance of subjectivity classification is that it could be used as an input data preprocessing step for sentiment classification. By filtering out objective sentences in advance of sentiment classification, subjectivity classification can increase the accuracy of sentiment classification.

## Sentiment Classification

Sentiment Classification is the process that aims to determine the sentiment orientation of a document, or part of a document. In other words, the objective of this phase is to classify each document or document segment into two different categories, *Positive* or *Negative*.

As with subjectivity classification, sentiment classification can be studied at different levels in a document, depending on the mining objectives. In general, each author focuses on one particular classification level, but a mixed approach is also possible. There is an extensive body of work on this topic, and a lot of techniques have been developed so far. A compilation of the most important techniques can be found in [62]. Techniques can be classified into three main groups:

- A. Classification Based on Supervised Learning: In general, any supervised ML method can be applied for this task, the most used ones being Naïve Bayes and Support Vector Machine. Features used by these methods include lexicons in conjunction with a set of rules and the result of POS tagging or parsing. Some of these rules and other features, including opinion lexicons, are explained below.
- B. Classification Based on Unsupervised Learning: The same features used in supervised learning can also be used in an unsupervised manner. For instance, a method proposed by Turney in 2002 [63] used some fixed syntactic phrases frequently used to give opinions in order to determine the sentiment orientation. On the other hand, the algorithm proposed by Popescu and Etzioni in [48] used POS tagging and similarity measures based on the pointwise mutual information (PMI) metric.
- C. Rule Based: On the other hand, instead of using a standard machine learning method, researchers have also proposed several custom techniques specifically for sentiment classification, like score functions and aggregation methods. The proposal of Ding, Liu and Yu in [64] is a good example of these techniques, achieving good results in general.

Some of the most common features used in machine learning are shown below:

- **Opinion Lexicon:** lexicon-based sentiment prediction is very popular. It generally relies on a sentiment word dictionary or so-called opinion lexicon. The lexicon typically contains a list of positive and negative words that are used to match words in the opinion text. Words contained in these lists are called *opinion words*, because they are commonly used to express positive or negative sentiments. In general, although many opinion words are adjectives and adverbs, nouns and other word categories can also indicate opinions, so lexicons can contain a wide variety of POS tags. Literature offers different options to develop opinion lexicons. Below, the three most common approaches, according to Liu in [37], are mentioned.
  - **Manual lexicon:** It is possible to manually compile a list of words with known orientations. Nevertheless, this method is very time-consuming. For that reason, in general, this method is not applied alone, but rather combined with any automatic technique.
  - **Dictionary-based lexicon:** This process starts with a small list of words with known orientations. Bootstrapping is applied to the list, using an on-line dictionary, like WordNet. *Bootstrapping* is a human language acquisition theory that proposes that human beings learn how to talk based on a small initial vocabulary which begins to grow by adding new words whose meaning is related to any word in the initial list. In the case of building *opinion words* lists, antonyms and synonyms are used to make the initial set of words grow. Several lists have been produced so far, Liu counts at least 5 in [37]. The main drawback of this approach is that it does not consider context or domain-dependent *opinion words*. Hu and Liu use this approach in [65].
  - **Corpus and sentiment consistency-based lexicon:** Starting with a small set of initial words, co-occurrence term patterns in a given corpus can be used to make the initial set grow. Nevertheless, the use of this technique alone does not assure as good effectiveness as the dictionary method, but is context and domain dependent.

It is important to consider that apart from individual words, there are also opinion phrases and idioms that might be indicating orientation. These phrases are instrumental to sentiment analysis for obvious reasons. After these words are found in text, the problem is how to combine them to determine the orientation of a document or a document segment. Approaches commonly include the application of Machine Learning Algorithms, which determine the orientation based on features appearing in text, besides *opinion words*.

- **Terms Frequency:** Individual words or word n-grams and their frequency counts are also commonly used. In some cases, word positions may also be considered. These features have been shown to be quite effective in sentiment classification.
- **Part of speech:** It was found in many studies that adjectives are important indicators of opinions. Thus, adjectives have been treated as special features.
- **NLP Rules:** Some special words (not included in the lexicon) appearing in the text might also change its orientation by following some intuitive linguistic rules. For instance, clearly negation words change the opinion orientation. Nevertheless despite how simple these rules might appear, it is important to handle them with care, because

not all occurrences of such rules or word apparitions will always mean the same. In this context, the rules developed by Ding, Liu and Yu in [64] are probably the most comprehensive ones, also achieving great performance.

- **Syntactic dependency:** Dependency-based word features generated from parsing or dependency trees are also tried by several researchers.
- **Ontologies:** Some authors have included the use of ontologies to support polarity mining. For example, Chaovalit and Zhou [66] manually built an ontology for movie reviews and incorporated it into a classification task, improving the performance over a standard baseline. A similar one was done by Khin Phyu and Phyu Shein [67].

## Opinion Summarization and Summary Visualization

The goal of summarization is to help user digest the vast availability of opinions in an easy manner. This idea is based on the proposal, made by Hu and Liu in [65], that seems more reasonable to analyse a collection of opinions rather than one, because one opinion represents only the point of view of a single person, which is not sufficient for action. Also, due to the huge amount of data available on the web, it is not possible either to check opinions one by one, so finding a summarization method becomes even more important [68].

In most of the cases, studies need to analyse a large number of opinion holders. Common sense indicates that one opinion from a single holder is usually not sufficient for action. This idea naturally leads to the task of opinion summarization. Literature proposes some different approaches to summarize and then visualize summarized opinions. This section focuses on different visualization techniques which require, in one way or another, some kind of previous summarization. This last task is a complex and pretty well-studied field. Its application to opinions (and web opinions) is just a particular case and will be briefly discussed in this section.

As presented in [69], traditional summarization consists of constructing new sentences from the opinionated document, in order to extract the document main points. In [69], the opinion summarization technique proposed is founded on the idea of analyzing relationships among basic opinionated units within the document. More precisely, the paper presents an approach to multi-perspective question answering (MPQA) that views the task as one of opinion-oriented information extraction. Briefly, the information extraction system takes as input an unrestricted text and summarizes the text with respect to a previously-specified topic or domain of interest, finding useful information about the domain and encoding that information in a structured form, suitable for populating databases. The process involves creating low-level annotations of the text which are then used to build the summary. Visualization in this context is thus the construction and presentation of short sentences (or sets of sentences) that capture a document's main opinionated ideas.

Traditional fashion for summarization then means producing a short text summary that gives the reader a quick overview of what people think about a defined object. Some traditional summarization techniques can be found in [70], [71], [72] and [73]. Nevertheless, the main weakness of these text-based summaries is that they are just qualitative, which means that it is not possible to apply any numerical or quantitative analysis to them. As proposed

in [37], the quantitative side is crucial, just as in traditional survey research.

The simplest form of a quantitative opinion summary is the result prediction, by aggregating the sentiment scores. Hence, some summarization techniques depend directly on the sentiment classification granularity. In this context, opinion quintuples defined by Liu’s approach are a good source of information for generating both qualitative and quantitative summaries, and can be stored in database tables. Based on this, a kind of summary based on aspects is defined, which is called aspect-based opinion summary [65], [39]. Having built the proposed structure, a whole set of visualization tools can be applied to see the results in all kinds of ways, to then gain insights into the opinions. In this case, bar charts or pie charts are both used. As an example, data can be visualized using a bar chart in which each bar above the X-axis shows the number of positive opinions on one aspect, and the corresponding bar below the X-axis shows the number of negative opinions on the same aspect. Figure 2.7 shows an example bar charts proposed by Liu.

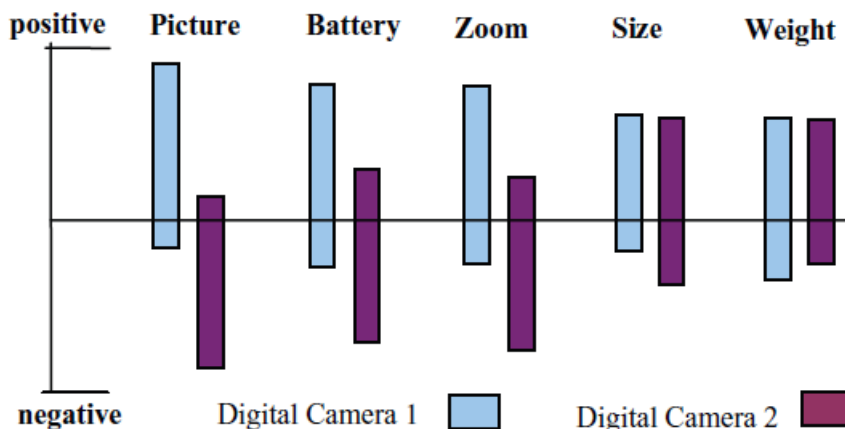


Figure 2.7: Bar chart showing a visual comparison of opinions on two products.

Source: Figure 2 in [74].

Liu’s visualization proposal is also interesting because it enables comparison of opinion summaries of some competing products. In addition to this, instead of just generating a quantitative summarization, a text summary directly from input reviews is also possible, generating natural language sentences based on what is shown in the charts [75].

It’s important to mention that this technique is only related to product opinions and results are quite different from traditional text summarization because it focuses and mines only the features of these products, while also determining whether the opinions are positive or negative. There is no rewriting of original sentences to capture the main points of the opinionated selected document, as in the classic text summarization described previously.

On the other hand, a completely different approach in relation to text summarization and visualization is presented in [38]. As presented before, traditional summarization algorithms rely on the important facts of opinionated documents and remove redundant information. Nevertheless, it’s likely that sentiment degree and correlated events play major roles in summarization. Because of that, [38] proposes that repeated opinions of the same polarity cannot be dropped because they strengthen the sentiment degree, but repeated reasons why they

stated a position should be removed when generating summarization. To apply this summarization system it was therefore needed to know which sentences were opinionated and then decide if they focused on a designated topic. An algorithm that detects and extracts major topics in long documents and then classifies them in positive or negative orientation in relation to that topic was then developed. Then, for brief summarization, the document with the largest number of positive or negative sentences is picked up and its headline is used to represent the overall tendency of positive-topical or negative-topical sentences. For detailed summarization, a list of positive-topical and negative-topical sentences with higher sentiment degree is generated. This type of summarization is classified as text selection in [46]. While statistical or quantitative summaries help users understand the overall idea of people’s opinion, text selection summaries are useful because sometimes, reading actual text is necessary to understand specifics. Since a large volume of opinions are usually collected on one topic, showing a complete list of sentences is not very useful. That is why many studies, including [38], [76], [48] and [47] try to show smaller pieces of text as the summary. They use different granularities of summaries including word, phrase and sentences level granularities.

As a complement an opinion-tracking system that shows how opinions change over time is proposed by Ku et al. in [38]. The tracking system is very similar to Liu’s proposal and consists of bar charts that simply count the number of positive-topical and negative-topical sentences on a selected topic at different time intervals. Nevertheless, a large number of relevant articles is required. Figure 2.8 shows an example of this time line-like summary.

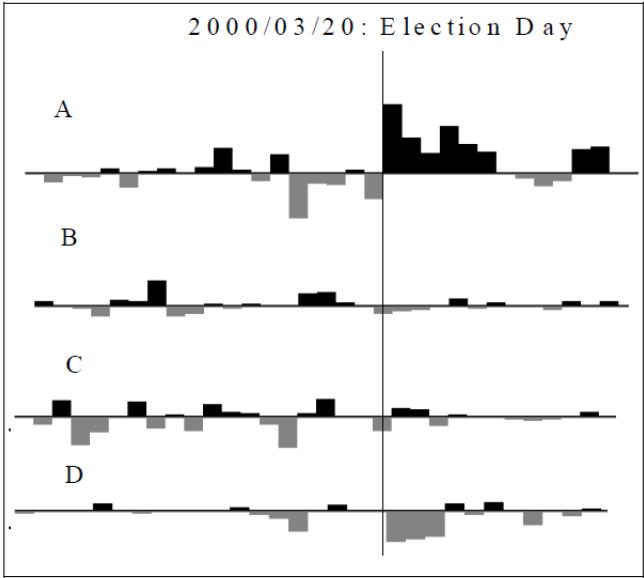


Figure 2.8: Opinions towards four persons in March 2000, during the president election in Taiwan. Candidate A was elected.  
Source: Figure 1 in [38]

Finally, Tateishi’s approach is worth mentioning, which introduces radar charts for summarizing opinions and has been frequently cited in the literature. A more detailed description of this technique can be found in [77]. Sadly, as the original document is only available in Japanese, it was not possible to include a deeper analysis due to language issues.



# Chapter 3

## Opinions and preferences

This chapter presents the ideas and models that will be used to fulfill the proposed objectives. In the first place, some important insights about preferences are given as an introduction. Also, some ideas about this term that are related to the conception of the models will be mentioned.

Later, the general opinion model is introduced, which will be the basis for all further work. This model was inspired by Bing Liu's work. Nevertheless, since it lacked some specifications that are important when applying it to extract real opinions, some additional propositions were defined. These propositions led to the development of a new model for extracting opinions from documents that is also presented.

Afterwards, methods and algorithms that will be used to extract opinions, according to the proposed models, will be explained and discussed in detail. These algorithms present a step-by-step procedure to obtain opinions from a set of documents.

Finally, it will be explained how the resulting opinion model is applied to finding consumer preferences. Consequently, it will be specified how an application that implements algorithms to process opinions constitutes a fairly promising alternative to finding these preferences. Also, some papers that have already been working on this topic will be mentioned.

### 3.1 Initial Concepts

In Chapter 1, it was established that the final objective of this work is to develop an Opinion Mining Application to find consumers preferences about tourism products in the X Región de Los Lagos. In the first place, the meaning of the concept *preference* needs to be clarified here.

Preference is a concept widely used in many scientific research areas, such as psychology, social sciences, economics and marketing. Looking from the point of view of each research field, the meaning of the concept *preference* could be slightly different. However, since

preferences have probably been introduced in scientific research by psychology, this field offers a quite comprehensive definition of it, stating that preferences could be conceived of as an individual's attitude towards a set of objects, typically reflected in an explicit decision-making process [78]. In this manner, as the most typical definition employed in psychology declares, one could interpret the term preference to mean evaluative judgement in the sense of liking or disliking an object [79].

Preferences are almost always assumed to have structural properties of a type that is best described in a formalized language. In fact, the study of the structural properties of preferences can be traced back to Aristotle. However, it was not until 1957 and 1963, that the first complete systems of preferences in math were proposed by Halldén and von Wright [80]. Since then, preferences started to be used in scientific fields that were more directly related to mathematics.

In economics, preferences have always been related to the process of customers choosing products or services. In demand analysis, the neoclassical consumer theory specifies that consumption goods can be selected by a consuming household by maximizing their utility function subject to their budgets [81]. Nevertheless, this static neoclassical model of consumer choice was later extended to accommodate taste change, introduction of new goods, and changes in the characteristics of the available goods. In 1966, Kelvin Lancaster revolutionized the world of economics with the development of what he called a *new theory of consumer demand*. In this theory, the standard microeconomic demand theory was modified by stating that consumer behavior is a process of choosing bundles of product characteristics or attributes inherent in goods and services, rather than simply choosing bundles of goods or services themselves [82]. In this manner, the product attributes model supposes that the consumer's choice is based on maximizing utility from the product attributes subject to a budget constraint. In other words, Lancaster proposed that preferences about products are not related to products as whole objects, but to a set of characteristics or features that products have. Subsequently, customers select products having features that maximize their wellness, obviously restricted to their budget.

In marketing, preferences appeared in a little different way. As proposed by Bartels in [83], there are some fundamental development stages of marketing thought regarding its history. A Period of Discovery, during the early 1900s, where initial teachers sought facts about the distributive trades (borrowing theory from economics), marked the conception of marketing. After some decades of development, the Periods of Reappraisal and Reconception appear (1940-1960). Over these years, the scientific aspects of the subject were considered and traditional approaches to the study of marketing were supplemented by increasing emphasis upon managerial decision making, the societal aspects of marketing and quantitative marketing analysis. Many new concepts, some borrowed from the field of management and from other social sciences, were introduced into marketing.

Thus, when marketing researchers ran into the product attributes model, a few years after it appeared, it immediately started to be used in applications, since it provided a robust way of predicting demand for products (also for new products) based on the characteristics or features they contained. Since then, the basic relevance of consumer preferences has been widely confirmed in marketing research and practice. Typically, consumer preferences are

estimated by means of conjoint analysis, using on-line or paper-and-pencil surveys. However, this type of preference elicitation can easily become expensive in terms of time and money. Moreover, the quality of the data resulting from consumer surveys directly depends on the willingness of the respondents to participate in the particular study and the length or complexity of the questionnaire [51]. Because of that reason, it seems worthwhile to consider alternative possibilities for determining and measuring consumer preferences.

Particularly, as seen in Chapter 1, when analyzing tourism demand by trying to discover consumer preferences using classical survey-based techniques, some of the mentioned issues are present. This study offers a first approach to considering a new alternative for discovering consumer preferences about tourism products using opinions available on the web.

Opinions have been playing an increasingly important role in the last few years since when a user gives an opinion on a product or service, the perception he transmits may contain really valuable information about his preferences. Opinion mining appears to offer a set of techniques to deal with all this data about opinions, delivering ways of obtaining valuable information about them. Particularly, as one simple opinion is surely not enough to generalize how consumers feel about a product or service, the special task of opinion summarization (one of the sub-tasks in opinion mining) has stirred tremendous interest among NLP and Text Mining communities around the world. Opinion summarization permits the simultaneous analysis of a huge number of opinions in little time (and without having to read every opinion), which may lead to a better understanding of consumer needs.

Nevertheless, before any opinion summarization attempt is developed, first it is needed to define what an opinion is. In opinion mining, literature offers two different main approaches to define and analyze opinions: at the document level or aspect-based opinion mining. While the last idea supposes that opinions are given about an entity, the entity subcomponents and attributes being the minimal opinion containers (in other words, considering text-segment granularity,) the former does not divide input text into subcomponents, usually limiting the analysis to a whole document as the minimal opinion container.

Given this scenario, an aspect-based approach has been selected in this work, since it permits analyzing opinions about product features, namely, product components and attributes. As established in Lancaster's model, customer preferences about a product are intrinsically related to its features. Thus, discovering what these features are and defining how customers feel about these features may undoubtedly lead to a better comprehension of tourism demand in the X Región de Los Lagos.

## 3.2 Models

In Chapter 2, it was shown that Bing Liu's aspect-based opinion mining approach focuses on determining what aspects are evaluated in opinions and what users tell about these aspects. The model is the result of several years of development and research. While his first work in relation to opinion mining appeared in the early 2000s and were almost purely based on classic data mining techniques, his last publications, dated last year (2012), show much more complex techniques, including advanced NLP and other linguistic tools.

Research showed that his work is probably the most comprehensive in aspect-based opinion mining, offering a complete range of techniques for each mining step he proposed. For that reason, the opinion mining model used in this study takes his ideas as inspiration. However, some original definitions and assumptions are analyzed and modified to fit this problem context.

Before any further explanation, bearing in mind that the chosen Opinion Model used in this work is basically Liu's and no big developments are going to be made in relation to it, it seems important to make clear what expected contributions this work aims to achieve:

- In the first place, using the model and implementing all proposed relevant steps in a completely different new domain (tourism) is presented as an important contribution, considering that so far, it has only been tested in analyzing opinions about physical products, like mobile phones, DVD players, photo cameras and others. In the same context, using all the tools the model provides to solve a business problem (finding customer preferences about tourism products in Los Lagos), also appears as a relevant contribution, validating how useful (or useless) the model is with real problems.
- Secondly, evaluating how the model behaves and performs in this new domain, studying how assumptions apply in this case and proposing possible new directions on research in this topic, emerge as another relevant contribution. In relation to this, as will be proposed later, the elaboration of tagged tourism opinion corpora becomes crucial.
- Finally, and perhaps more importantly, certain ambiguities were found during research or implementation, either in the original definitions or the process steps proposed by Liu. In some cases, these ambiguities led to relatively important issues which needed to be solved in order to achieve some of the project goals. The contribution here is that all ambiguities that were found are discussed and solutions are proposed for them considering a linguistic point of view, always trying to keep assumptions as simple or real as possible, and hopefully setting some common ground for future work.

### 3.2.1 Opinion Model

Let's now move on to explaining the model, which was taken from [37]. It considers opinions as objects composed of 5 components, as follows.

- *Entity*: Without loss of generality, it can be stated that opinions are expressed about something, like a product, service, organization, event, etcetera. Therefore, the concept of *entity* is proposed to denote the opinion objective or, in other words, what is being evaluated by the opinion. In this manner, an *entity* can contain a set of components and, at the same time, a set of attributes. Similarly, each *entity* component can have its own subcomponents and attributes, so finally, an *entity* can be decomposed into a tree or hierarchy.
- *Aspect*: Because it is difficult to study an *entity* at an arbitrary hierarchy level, this hierarchy is simplified to one or two levels, denoting *aspect* every component or attribute of the *entity*. In this way, the root of the hierarchy or tree becomes the *entity* itself,

each leaf is an *aspect* and links are part-of relations.

- *Sentiment Orientation*: Basically, opinions express a *positive* or *negative* sentiment about what they evaluate (in some cases, *neutral* sentiment is also considered, but it is also sometimes considered as no opinion). Given these opinion types or classes, the *sentiment orientation* of an opinion is the result of classifying it into one of these classes. Even though in practice it is possible to determine the *sentiment degree* of an opinion, the proposed model simply measures the standard opinion *sentiment orientation*.
- *Opinion Holder*: Corresponds to the user (a person, an enterprise, etc.) that gives the opinion. Most of the time, when dealing with opinions available on websites, the user is usually the comment or review author, as seen in blogs and other similar web opinion sources.
- *Time*: Time and date (time only if available) when the opinion was given.

Ultimately, opinions are simply a positive or negative view, attitude, emotion or appraisal about an entity or an aspect of that entity from an opinion holder. Using mathematical notation and given  $D$ , a set of documents with opinions, opinions are defined as 5-tuples:

$$(e_i, a_{ij}, oo_{ijkl}, h_k, t_l) \quad (3.1)$$

Where:

$e_i$ : *Entity* name  $i \in I$ , set of *entities* in  $D$ .

$a_{ij}$ : *Aspect*  $j$  of *entity*  $i$ .  $j \in J(i)$  set of *entity*  $e_i$  *aspects*.

$oo_{ijkl}$ : *Sentiment orientation* of the opinion about *aspect*  $j$  of *entity*  $i$ , given by *opinion holder*  $k$  in *time*  $l$ .  $oo_{ijkl} \in \{Positive, Negative\}$ .

$h_k$ : *Opinion holder*  $k \in K$ , set of *opinion holders* in  $D$ .

$t_l$ : *Time* when the opinion is expressed, with  $l \in L$  set of *times* when opinions in  $D$  were given.

In the model, these opinion tuples present 2 basic properties:

1. The five pieces or components that compose the tuple must correspond with/to one another. In other words, the opinion  $oo_{ijkl}$  must be given by holder  $h_k$ , about aspect  $a_{ij}$  of entity  $e_i$ , in time  $t_l$ .
2. The five components are essential and without any of them, opinions tend to lose sense. Nevertheless, for a particular application, all five components might not strictly be needed. Also, extra components could be added if necessary or interesting.

This abstraction allows seeing a structure within complex unstructured text. Generating this structure permits the performance of qualitative and quantitative analysis of opinions, since quintuples give specific information necessary to create, for instance, an indexed database containing opinions, which is a lot easier to manage and process than text. However,

although the model seems to be a solid framework to analyze opinions, two more important concepts are needed. These concepts are related to the fact that users could use multiple terms to refer the same aspect or entity in their opinions. Therefore, the following concepts are introduced:

*Entity Expression*: Corresponds to the actual word or phrase written by the user to denote or indicate an entity. As a result, entities are then generalizations of every entity expression used in the analyzed documents, or a particular realization of an entity expression. In [37], this concept is called entity name.

*Aspect Expression*: As for an *entity expression*, the aspect expression is the actual word or phrase written by the user to denote or indicate an aspect. Thus, aspects are also general concepts that comprise every aspect expression. They are called aspect names by Bing Liu.

Having defined these concepts, it is then possible to define a model of an entity and a model of an opinionated document, finally making clear what the mining objective is.

- Model of *entity*: An *entity*  $e_i$  is represented by itself as a whole and a finite set of aspects,  $A_i = \{a_{i1}, a_{i2}, \dots, a_{in}\}$ . The *entity* itself can be expressed with any one of a final set of entity expressions  $EE_i = \{ee_{i1}, ee_{i2}, \dots, ee_{is}\}$ . Each *aspect*  $a_{ij}$  of  $A_i$  of *entity*  $e_i$  can be expressed by any one of a finite set of *aspect expressions*  $AE_{ij} = \{ae_{ij1}, ae_{ij2}, \dots, ae_{ijm}\}$ .
- Model of opinionated document: An opinionated document  $d_k \in D$  contains opinions on a set of *entities*  $e_1, e_2, \dots, e_r$  from a set of opinion holders  $h_1, h_2, \dots, h_p$ . The opinions on each *entity*  $e_i$  are expressed on the *entity* itself and a subset  $A_{ik}$  of its *aspects*.

Then, the objective of the Aspect-Based Opinion Mining approach is to, given a set of opinionated documents  $D$  (a set of documents that contain opinions), discover all the opinion tuples in  $D$ .

### 3.2.2 Opinion Extraction Model

Modeling involves abstraction, simplification and formalization, in light of particular methods and assumptions, in order to better understand a particular part or feature of the world, and to potentially intervene [84], [85].

Human language is unique in comparison to other forms of communication [86], [87]. It allows humans to produce an infinite set of utterances from a finite set of elements, and to create new words and sentences. This is possible because human language is based on a dual code, where a finite number of meaningless elements (including sounds, letters or gestures) can be combined to form units of meaning (words and sentences) [88]. This special condition of human language is probably what makes developing a model that explains everything about language a very, very complex task.

Nevertheless, in scientific research, models do not pursue perfection. In fact, scientific models seek to represent empirical objects, phenomena, and physical processes in a logical

and objective way. All models are in simulacra, that is, they are simplified reflections of reality, but, despite their inherent falsity, they are nevertheless extremely useful [89].

This short reflection about how models work and how they should be understood leads to a better discussion about the Opinion Model introduced, bearing in mind that, language is much more complex than what the model proposes.

In first place, orientation is not only positivity/negativity about something. Indeed, perceptions or the position someone could have about something is not even remotely binary; language is a lot more than positive or negative appreciations about something.

On the other hand, reducing opinions to be given in only one single word (the *aspect* or *entity* in a sentence) relies on a very reductionist principle. In text linguistics, discursive or speech genres try to determine what characteristics make a text belong to a specific type. According to Mikhail Bajtin, who made one of the best theorizations about discursive genres, a speech genre is a set of stable statements that can be grouped as having similarities in their thematic content, verbal style and composition [90]. One of the most important conclusions that can be obtained from this proposal is the fact that speech genres define realities, namely, prototypical ways of communicating ideas. Society defines these realities by generating conventions, that are somewhat implicit, and not by using rules. That is why studying language from a computational point of view turns out to be so complex and frequently so limited in its interpretations. Indeed, from a strictly linguistic point perspective, there are still no descriptions of structural characteristics that an opinion or review should have. Despite that, in [91], Pollach states that consumer web opinions can be considered a truly digital genre, as consumers were not able to share their opinions with other consumers in a structured, written format before the advent of Internet. So, anyway, there is still a lot of work to do in this field.

However, considering that it is possible to define a structure in opinions, common sense dictates that there might be types of opinions. For instances, regular opinions (the ones that have been discussed above) are different from comparative opinions, which often tend to express similarities and differences between two or more entities or aspects of those entities. These comparative opinions can be easily recognized as they are usually expressed using the comparative or superlative form of an adjective or adverb.

As if that was not enough, there might also be sub-types of regular opinions. For instance, it is possible to define direct opinions as those that are expressed about entities or aspects, while indirect opinions would consider all other kinds of opinions, like the ones that are expressed based on the effects of entities or aspect on some other entities. These last kinds of opinions are frequently found in the medical domain [37].

Last, but not least, due to the simplification the model applies on the entity hierarchy tree by flattening it using aspects to denote attributes and components in the same way, opinion quintuples can result in important information loss. This issue becomes important in cases where opinions about different aspects of an entity are to be studied. In this case, the quintuple model applies anyway, but it would need to include the part-of relationship between an aspect and their respective entity, obviously making the problem more complex

by unflattening the hierarchy tree and understanding opinion tuples as nested relationships. Another simpler solution perspective is considering each aspect as a separate entity.

The discovery and construction of these opinion tuples is achieved by following a set of structured steps. The design of these steps has been developed by Bing Liu in several publications since he began working on Opinion Mining. Because of that reason, in some cases there are different alternatives that are capable of achieving one step objective. Some other authors have also proposed their own alternatives. Nevertheless, during the analysis of Liu’s propositions available in literature, some inconsistencies between what these techniques propose and the opinion model developed below were found. All these inconsistencies come to light when the model is applied to real data. In other words, the model lacks important definitions (or assumptions) that are needed when trying to obtain opinion tuples from real opinionated documents.

In the first place, as stated above, *aspect expressions* are used in texts to indicate *aspects*. Two main reasons explain the fact that many expressions could indicate the same concept:

- The economy principle in languages [92] restricts the amount of elements that can be used in communication to the least possible, so that the instrument (language) is flexible and does not require too many compositional elements. In simple words, languages try to say a lot using few words. For example, the sentence *The hotel has good wifi*” corresponds to a lexicalization, where the original expression should have been *The hotel has good internet access through wifi* is shortened according to the economy principle.
- Each language presents systems that organize its concepts, also pursuing simplification. For that reason, many words in English (as in all other languages) simply are hyponyms of a determined hypernym. An hyponym is a word or phrase whose semantic field is included within that of another word, its hypernym. For instances, scarlet, vermilion, carmine, and crimson are all hyponyms of red (their hypernym), which is, in turn, a hyponym of color [93].

Then, in practice, finding the *aspects* that are evaluated in a set of opinionated documents is a really complex task. In fact, detecting *aspect expressions* from a set of documents with opinions should be a completely different task than defining or finding the real *aspects* in them.

On the other hand, it is also unclear how aspects that appear more than once in a document are managed in building tuples. Consequently, a limit for determining semantic orientation about aspects is also unknown. In other words, the model fails to clearly specify the text granularity to consider when constructing tuples from an opinionated document. Having noticed this important issue, a model to build opinion tuples from an opinionated document has been developed.

Consider a set of opinionated documents about only one entity,  $e_i$ ,  $D_i = \{d_{11}, d_{12}, \dots, d_{1m}\}$ . Each opinionated document will correspond to a review or opinion given by holder  $h_k$  in time  $t_k$ , Let  $S_{ik}$  be the set of all sentences in  $d_{ik}$ , with  $S_{ik} = \{s_{ij1}, s_{ij2}, \dots, s_{ijn}\}$ . Then, opinions on



$e_i$  in  $d_{ik}$  are expressed on the *entity* itself and on a subset  $A_{ik}$  of its *aspects*. Similarly, each aspect of  $A_{ik}$  will appear on  $d_{ik}$  as a set of aspect expressions  $AE_{ijk}$ , subset of  $AE_{ij}$ .

The set  $AE_{D_i}$  will be defined as the set of all *aspect expressions* of all *aspects* and all *entity expressions* appearing in  $D_i$ . Using mathematical notation:

$$AE_{D_i} = \left[ \bigcup_{k=1}^m \left[ \bigcup_{AE_{ijk} \in A_{ik}} AE_{ijk} \right] \right] \cup \left[ \bigcup_{k=1}^m EE_{ik} \right] \quad (3.2)$$

Then, the binary relation  $R$ , between elements from sets  $AE_{D_i}$  and  $S_{ik}$ , is defined as follows:

$$R = \{(ae, s) : ae \in AE_{D_i} \wedge s \in S_{ik} \wedge ae \in s\} \quad (3.3)$$

In other words, sentence  $s_{ikl} \in S_{ik}$  is related to one *aspect expression*  $ae_{ir} \in AE_{D_i}$  only if it appears in sentence  $s_{ikl}$ . Then, sentiment orientation needs to be determined for each pair  $(ae, s)$  only if  $aeRs$ . After determining sentiment orientation,  $h_k$  and  $t_k$  of the corresponding document  $d_{ik}$  should simply be added in order to build a complete opinion tuple. From this, it follows that each sentence  $s|\exists ae, aeRs$ , will have an associated opinion tuple.

$$(e_i, ae_{ir}, oo_{irkl}, h_k, t_k) \text{ with } ae_{ir}Rs_{ikl} \quad (3.4)$$

With:

- entity  $i$ :  $e_i$
- *aspect expression*:  $ae_{ir}$  in  $AE_{D_i}$
- orientation:  $oo_{irkl} \in \{Positive, Negative\}$
- opinion holder:  $h_k$
- time:  $t_k$

### 3.3 Opinion Mining Algorithms

In this section, algorithms selected to achieve what the opinion extraction model proposes are explained in detail. However, some initial definitions that will be used in the algorithms are needed first.

Although sentences are one of the most basic structures in text, giving a definition of what a sentence is could be quite complex. Despite linguistic analysis, here a sentence will be considered as an ordered set of tokens, including words and punctuation. One token that appears in two different positions must be considered twice, as the positions where

they appear are distinct. In other words, a sentence  $S$  will correspond to a set of unique tuples (*token, position*). According to this, for instance, the sentence “*My brother and I like basketball, but he is really bad at it.*” would look like this:

$[My]$   $[brother]$   $[and]$   $[I]$   $[like]$   $[basketball]$   $[,]$   $[but]$   $[he]$   $[is]$   $[really]$   $[bad]$   $[at]$   $[it]$   $[.]$   
 0            1            2            3            4            5            6            7            8            9            10            11            12            13            14

In the example, token positions start with 0, but in practice this is arbitrary. Positions can only be in  $\mathbb{N} \cup \{0\}$  and the difference between two adjacent components must be 1.

Given this sentence model, the concept of *word distance* between two elements of sentence  $S$  will correspond to the difference of the positions of the two tokens in  $S$ .

$$\text{Word Distance}(t_a, t_b) = |\text{position}(t_a) - \text{position}(t_b)| \text{ with } t_a, t_b \in S \quad (3.5)$$

As  $\text{Word Distance}(t_a, t_b)$  is simply the absolute value of the difference between numbers in  $\mathbb{N} \cup \{0\}$ ,  $\text{Word Distance}(t_a, t_b)$  is a metric on the set  $S$  as it satisfies the conditions of non-negativity, identity of indiscernibles, symmetry and triangle inequality. Note that minimal distance between 2 elements in  $S$  is 1, and it occurs between adjacent elements. The maximum distance corresponds to  $|S| + 1$ .

It is important to mention that these concepts, although used by Bing Liu in the algorithms he proposes, have not been clearly defined as far as it is known. That is the reason why they were defined here. Nevertheless, these definitions are far from being the best possible for all situations and should be considered as one first approach in defining a set of robust tools in NLP. Indeed, one important detail refers to the fact that sentences are one complex structure in linguistics and recognizing them in text is not always straightforward.

### 3.3.1 *Aspect expression extraction*

The extraction of explicit *aspect expressions* was one of Bing Liu’s first publications, and most of the initial framework can be found in [49]. His work proposes that since aspect expressions are usually nouns and noun phrases, when people comment on different aspects of a product, the vocabulary that they use usually converges. Thus, nouns that are frequently talked about are usually genuine and important *aspect expressions*, and should be recognized as *aspect candidates*.

Nevertheless, it is important to keep in mind that *aspect expressions* could also be verbs, verb phrases, adjectives, adverbs and also idiomatic expressions. Then, *aspect expressions* that appear in a sentence as nouns and noun phrases are here called *explicit aspect expressions*. On the other hand, *aspect expressions* of all the other types are called *implicit aspect expressions*, since they can imply or indicate *aspects* indirectly.

For example, in “*This hotel room was good.*”, “*room*” is an *explicit aspect expression*. But, for instance, “*small*” is an *implicit aspect expression* in the sentence “*The room they gave*

*us was really small.*”, since it implies the noun “*size*”. As the last example shows, there are many *implicit aspect expressions* that are adjectives, which often imply some specific *aspects* as nouns.

*Aspect expressions* are usually nouns because in most languages the noun category includes words denoting all kinds of physical objects (people, animals, places, things) and substances. However, this criterion can’t be used for identifying English nouns, because there are also large numbers of nouns denoting abstract entities [94].

Considering all the ideas above, the following steps are proposed:

## 1. Preprocessing

- (a) Tokenize each opinionated document, obtaining the sentences that compose it.
- (b) For each sentence in each document, tokenize again, to obtain the sets of words in each sentence.
- (c) Apply POS tagging to the tokens of each sentence (tokenization was applied because, in general, POS tagging processes take tokens as input).
- (d) Apply a process of chunking to the sentences, to identify noun and noun phrases (NPs). In most cases this step requires POS-tagged sentences (a result of the previous step).
- (e) Store each opinion sentence in a database, including POS tags, and marked nouns and NPs (Bing Liu stores the data using XML files).
- (f) Create a transactional file (Bing Liu uses a plain text file) to identify *frequent aspects* (see step 2). In this file, each line must contain only the nouns and the NPs of a sentence, all written in lower case. A line is written for each sentence. Other sentence components are deleted as they are not likely to be *aspects*, following the steps below.
- (g) Apply a stopwords deletion to the file, in order to eliminate words that were part of NPs but do not add valuable information.
- (h) Application of stemming algorithms to the file.
- (i) Application of a fuzzy matching process to the file, the purpose of which is to handle typos, spelling mistakes and concepts that might be the same, but have been written in a little different manner.

These three last steps aim to reduce dimensionality of the transactional file, and eliminate nouns that might indicate some abstract entities, making the *aspect expression* extraction process easier to be run next.

## 2. Frequent aspects extraction

- (a) An association rules mining process is applied to the transactional file. This process aims to find the most frequent itemsets, in this case the most frequent sets of nouns and NPs. Under Liu’s criteria, itemsets that have a minimum support of 1% (i.e. appear in at least 1% of the file lines or sentences) and are composed of at most three words, are chosen as *frequent aspect candidates*. The 3-word maximum rule is imposed as it seems difficult to find longer aspects in practice.

- (b) Each *frequent aspect candidate* found is saved, marking as *simple candidates* those that are conformed by only one word, and as *compound candidates* those that are conformed by two or three words (or more).
3. **Compactness Pruning:** From the *compound candidates* list, those that might be nonsense should be removed. To assure that the words that compose the *compound aspect* are likely to represent a coherent object together, a new metric that gives the notion of a *compact aspect* is introduced:

*Compact Aspect Expression:* Let  $a$  be a *compound candidate* conformed by  $n$  words, with  $1 < n \leq 3$ . Also, let the sentence  $s$  contain  $a$  and the sequence of words that appears in  $a$  is  $p_1, p_2 \dots p_n$ . If in the sentence  $s$  the word distance between two adjacent words in  $a$ ,  $p_i, p_{i+1}$ ,  $i \in \{1 \dots n - 1\}$  is less than or equal to 3, then it is said that  $a$  is compact in  $s$ .

- (a) Then, for each *compound candidate*  $a$ , if it is not possible to find two sentences in which  $a$  is *compact*, the aspect  $a$  is pruned.
4. **Redundancy Pruning:** Redundant *simple candidates* need to be eliminated. To decide which *simple candidates* needs to be pruned, *p-support* is defined.

*P-support:* Let  $a$  be an arbitrary aspect, its *p-support* is the number of sentences in which  $a$  appears as a noun or NP, and these sentences mustn't contain any *compound candidates* that  $a$  are part of, i.e. whose words belong to  $a$ .

- (a) Those *aspect candidates* (*simple* or *compound*) that have a *p-support* of less than 3 and are subsets of other *compound aspects* are pruned.

As this process ends, a list with the *aspect candidates* is generated. Each aspect on this list will be called a *frequent aspect*.

### 5. *Infrequent aspects* extraction

- (a) For each sentence in the database, if it does not contain any *frequent aspects*, but it does contain one or more *opinion words/phrases* from the *opinion lexicon*, the *opinion word/phrase* nearest the noun or NP is extracted.
- (b) All nouns and NPs extracted in the previous step are added to the list of *infrequent aspects*.

Before moving on the next algorithm, it is important to mention that criteria proposed by the algorithm are purely empirical. Minimum support rules to extract frequent itemsets will directly depend on the size of the document that is being analyzed. Similarly, compactness rules and *p-support* rules could also change under certain conditions.

## 3.3.2 Determination of the Opinion Orientation

To accomplish this step, a dictionary-based lexicon approach is used. As mentioned in Chapter 2, several authors have developed their own dictionaries and most of them are publicly available on the web. In this case, Bing Liu's dictionary (a list of positive and negative words for English), with around 6,800 words, is used. This list was compiled over

many years and can be found at <http://www.cs.uic.edu/liub>. Every term that appears in this dictionary will be called *opinion word*, since they are words that usually qualify objects or attributes of objects. Opinion words are usually adjectives and adverbs, but they can also be nouns and verbs. Also, the list of *opinion words* used include many misspelled words, which are included as they appear frequently in social media content.

Before explaining how these *opinion words* will be used to determine orientation, it is important to state that the appearance of an opinion word in a sentence does not necessarily mean that the sentence expresses a positive or negative opinion. Similarly, the absence of these words does not necessarily imply that the sentence has no opinion. On the other hand, apart from opinion words, idioms could also indicate orientation. However, since public lists of positive and negative idioms could not be found, they will not be considered here. Also, at some point the possibility of generating the list was studied, because although this task can be time consuming, it is only a one-time effort. Finally, this idea was discarded as it was too time consuming.

Using a dictionary-based lexicon approach presents some key difficulties. They are:

1. How to combine multiple opinion words in a sentence (which may be conflicting) to arrive at the final decision of the orientation.
2. How to deal with language constructs which can change the semantic orientations of opinion words.
3. How to deal with context or domain dependent opinion words without any prior knowledge from the user.

Taking Liu's work in [64] as inspiration, a set of rules to determine the sentence orientation was developed, always considering *opinion words* as a basis. The point of these rules is to solve difficulties 1 and 2. Although a method of dealing with context or domain-dependent *opinion words* is proposed by the authors, this task will be left for future work as it could be quite complex in the tourism domain. Below, each rule is explained to finally present how all of them should be applied and combined in order to determine the final orientation.

## Word Orientation Rules

- **Opinion Word Rules:** Positive *opinion words* will intrinsically have a *score* of 1, denoting a normalized positive orientation, while negative ones will have associated a *score* of  $-1$ , denoting a normalized negative orientation.
- **Neutral Word Rules:** Every noun and adjective in each sentence that is not an *opinion word* will have an intrinsic *score* of 0 and will be called *neutral words*.
- **Negation Rules:** A negation word or phrase usually reverses the opinion expressed in a sentence. Consequently, *opinion words* or *neutral words* that are affected by negations need to be specially treated. Three rules must be applied:
  - Negation Negative ( $score = -1$ )  $\rightarrow$  Positive ( $score = +1$ )

- Negation Positive ( $score = +1$ )  $\rightarrow$  Negative ( $score = -1$ )
- Negation Neutral ( $score = 0$ )  $\rightarrow$  Negative ( $score = -1$ )

Negation words and phrases include: *no, not, never, n't, dont, cant, didnt, wouldnt, havent, shouldnt*. Also, some negation patterns are considered, including *stop + vb-ing, quit + vb-ing* and *cease + to + vb*.

- **Too Rules:** Sentences where words *too, excessively* or *overly* appear, are also handled specially. The following rules must be applied for *opinion words* or *neutral words* appearing near one of the mentioned terms, which will be denoted as *too words*. As taken from [64], this is considered since the presence of the *too word* in the text indicates that the user is not happy about an attribute.
  - Too word + Positive ( $score = +1$ )  $\rightarrow$  Negative ( $score = -1$ )
  - Too word + Negative ( $score = -1$ )  $\rightarrow$  Negative ( $score = -1$ )
  - Too word + Neutral ( $score = 0$ )  $\rightarrow$  Negative ( $score = -1$ )

## Aspect Orientation Rules

Having mentioned rules that help in determining each word orientation in a sentence, it is now explained how all these orientations should be combined to determine the final orientation of a sentence on a particular aspect. This algorithm should only consider words marked as *opinion words* or *neutral words* as they are the only ones that will provide an orientation for each sentence.

- **Aspect Words Aggregation Rule:** Let  $s$  be a sentence that contains the set of *aspect expressions*  $A = \{a_1, \dots, a_m\}$ , each one of them appearing only one time in  $s$ . Also, let  $AW_i$  be the set of words that compose aspect  $a_i$ , where  $AW_i = \{aw_{i1}, aw_{i2}, \dots, aw_{in}\}$ . Each  $aw_{ij}$  will be called *aspect word* and it will correspond to *aspect expression*  $a_i$ . If *scores* for each *opinion word* and *neutral word* in  $s$  are known, *score* for each  $aw_{ij}$  in  $s$  is given by the following aggregation function:

$$score(aw_{ij}, s) = \sum_{ow_j \in s} \frac{score(ow_j)}{Word\ Distance(ow_j, aw_{ij})} \quad (3.6)$$

Where  $ow_j$  is an *opinion word* or *neutral word*. In the formula, the multiplicative inverse is used to give low weights to opinion words that are far away from the *aspect expression word*. As explained in [64], far away *opinion words* may not modify the current feature. However, setting a distance range or limit within which the *opinion words* are considered does not perform well either because in some cases, the *opinion words* may be far away. The function deals with both problems nicely.

- **Aspect Aggregation Rule:** For each *compound aspect expression*  $a_i$  in  $s$ , its score will be calculated considering the *scores* of all the words that compose it,  $aw_{ij} \in AW_i$ , according to the following equation:

$$score(a_i, s) = \sum_{j \in A_i} score(aw_{ij}, s) \quad (3.7)$$

Where  $score(aw_{ij}, s)$  is the *score* of  $a_{ij}$  in sentence  $s$ . From the formula, it follows that in case the aspect  $a_i$  is composed by only one word  $wa_{i1}$ , its score will simply be the score of  $aw_{i1}$ .

- **Position Aggregation Rule:** Considering that an *aspect expression* could appear multiple times in one same sentence, a method for determining the final opinion orientation on that *aspect expression* is needed. Let  $a_i$  appear  $t$  times in  $s$ ,  $\{a_i^1, a_i^2, \dots, a_i^t\}$  and suppose that orientation about *aspect words* of all  $a_i^k, k \in \{1, 2, \dots, t\}$  is known. Aspect *score* will be calculated aggregating scores of each appearance, as follows:

$$fscore(a_i, s) = \sum_{k=1}^t score(a_i^k, s) \quad (3.8)$$

Where  $score(a_i^k, s)$  is the score of  $a_i^k$  in  $s$ . Note that in cases that the *aspect expression* appears only one time in the sentence, its final score will simply be the score of its only appearance in  $s$ . Finally, if the *score* of  $a_i$  in  $s$  is positive, the opinion is considered positive on  $a_i$  and if the final *score* is negative, the opinion is considered negative on  $a_i$ . If none of these cases occur, the opinion orientation is zero.

- **But Clauses Rules:** A sentence including the word *but* (or any synonym) needs special treatment. This occurs because frequently, opinions before and after *but* have opposite orientation. Then, let  $b$  be a *but word* appearing in sentence  $s$ . Also, let  $s$  contain the set of *aspects*  $A = \{a_1, \dots, a_m\}$ , where also  $AW_i = \{a_{i1}, a_{i2}, \dots, a_{in}\}$ . In that case, the following rules must be applied:
  - Break the  $s$  into two segments, the one before and the one after  $b$ . Consider each segment as a separate sentence.
  - If the orientation of any *aspect word*  $aw_{ij}$  appearing in the sentence segment after  $b$  is zero, its orientation should then be determined using the segment before  $b$  and negating it. Since in any of these cases  $a_{ij}$  could not truly be inside the considered segment, it must be added at the final position of the corresponding segment to avoid consistency issues.

This two rules suppose that for determining the orientation of each  $aw_{ij}$  in  $s$ , the orientation of the but clause is used first. If it is not possible to get an orientation from that clause, the clause before the but word is used. In case an orientation is not found in the clause before the but word, the final orientation of  $aw_{ij}$  in  $s$  will be zero.

Note that when the algorithm is applied to a sentence where there are no *opinion words* or *negations words* or *too words*, the resulting orientation on any aspect in that sentence will be 0. This could indicate that, although one aspect is mentioned in the sentence, no opinion is given on it (in which case, the orientation being 0 would be right), or even though an opinion has been given, the *opinion words* that were used in it are not on the list.

It is worth mentioning that Liu’s work did not consider that an *aspect expression* could appear more than once in a sentence. However, experience showed that it occurs very often. Also, the algorithm in [64] lacked an explanation about determining the orientation on *compound aspect expressions*. Considering these two issues, the idea of determining orientation on each *aspect expression word* and then combining those orientations to obtain the final result, was here proposed.

Finally, some presented rules do not cover all the cases. For instance, there might be non-negation containing negation words or non-but clauses containing but-like words (e.g. not only but also.) Such phrases should be identified and treated separately. The following algorithms summarize all the presented ideas.

---

**Algorithm 1** Opinion Orientation(aspect,marked\_words,sentence)

---

```

1: if but_word is in sentence then
2:   orientation  $\leftarrow$  Opinion Orientation(aspect,marked_words,but_clause)
3:   if orientation  $\neq$  0 then return orientation
4:   else
5:     orientation  $\leftarrow$  Opinion Orientation(aspect,marked_words,not but_clause)
6:     if orientation  $\neq$  0 then return -1  $\times$  orientation
7:     else
8:       return 0
9:     end if
10:  end if
11: else
12:  for all aspect_position in aspect do
13:    for all aspect_word in aspect_position do
14:      for all word in marked_words do
15:        suborientation  $+= \frac{Word\ Orientation(word)}{Word\ Distance(aspect\_word,word)}$ 
16:      end for
17:      orientation  $+=$  suborientation
18:    end for
19:    final_orientation  $+=$  orientation
20:  end for
21:  if final_orientation  $>$  0 then
22:    return 1
23:  else
24:    if final_orientation  $<$  0 then
25:      return -1
26:    else
27:      return 0
28:    end if
29:  end if
30: end if

```

---



---

**Algorithm 2** Word Orientation(marked\_word)

---

```
1: if marked_word is in opinion_words then
2:   orientation  $\leftarrow$  Apply Opinion Word Rule(marked_word)
3: else
4:   if marked_word is in neutral_words then
5:     orientation  $\leftarrow$  0
6:   end if
7: end if
8: if marked_word is near a too_word then
9:   orientation  $\leftarrow$  Apply Too Rules(orientation)
10: end if
11: if marked_word is near a negation_word then
12:   orientation  $\leftarrow$  Apply Negation Rules(orientation)
13: end if
14: return orientation
```

---

## 3.4 Finding customer preferences through opinions

In section 3.1, a model to extract opinions from opinionated documents was proposed. This model permits the generation of a well-defined structure from text. This structure, namely, the opinion tuples, could be used in different ways to help in understanding customer preferences. Here, some of these possible uses are proposed.

### 3.4.1 Using Aspect-Based Opinion Summary

Aspect-Based Opinion summaries offer a simple way to show all the opinion tuples that have been generated about one entity. A whole suite of database and visualization tools can be applied to see the results in all kinds of ways to gain insights into the opinions in structured forms and displayed as bar charts and/or pie charts [37]. Particularly, bar charts that show the number of positive and negative opinions about every aspect of the entity are proposed by Bing Liu in [75]. In this paper, Liu proposes that the bar charts could be used to compare a set of selected products, showing the set of all aspects of the chosen products in the chart. In this case, each bar above or below the x-axis can be displayed in two scales:

1. The actual number of positive or negative opinions normalized with the maximal number of opinions on any feature of any product (this is to ensure that the tallest bar fits the limited space.)
2. Percent of positive or negative opinions, showing the comparison in terms of percentages of positive and negative reviews.

This proposal seems fairly simple and effective. However, it lacks a robust way of measuring the importance of each evaluated *aspect*. In [65], *aspects* are ranked according to the frequency of their appearances in the reviews, but it is also declared that other types of rankings are also possible, like ranking *aspects* according to the number of reviews that express

positive or negative opinions.

Here, a new proposal was developed. The proposal is an attempt at measuring the importance of each *aspect* simultaneously using the amount of positive and negative opinions of it. The underlying assumption is that an *aspect* that has a lot of positive and negative opinions will more important, since the high number of opinions of both orientations might indicate that customers are very interested in that *aspect*. In this way, not only the total number of times that an *aspect* appears is considered in measuring importance, but also the dispersion in the number of positive and negative opinions.

Complementing the last proposal with Bing Liu's ideas, a new opinion summary is proposed. Suppose that a set of opinion tuples about entity  $e$  have been generated using the set of opinionated documents  $D$ . The set of extracted aspects of entity  $e$  will be  $A = \{a_1, a_2, \dots, a_n\}$ . Likewise, the set of extracted opinion holders will be  $H = \{h_1, h_2, \dots, h_m\}$  and the set of extracted times will be  $T = \{t_1, t_2, \dots, t_p\}$ . Finally,  $Pa_i$  and  $Na_i$  will be the set of positive and negative opinions on aspect  $a_i$ . Then, the following measures are defined for each aspect  $a_i \in A$ :

Positive Score: Number of positive opinions on  $a_i$  .

$$PScore_i = |Pa_i| \quad (3.9)$$

Negative Score = Number of negative opinions on  $a_i$ .

$$NScore_i = |Na_i| \quad (3.10)$$

Normalized Positive Score =

$$NPScore_i = \frac{PScore_i - MinScore}{MaxScore - MinScore} \quad (3.11)$$

Normalized Negative Score =

$$NNScore_i = \frac{NScore_i - MinScore}{MaxScore - MinScore} \quad (3.12)$$

Relative Importance =

$$RI_i = \frac{STDScore_i - \min_{i \in \{1, \dots, n\}} STDScore_i}{\max_{i \in \{1, \dots, n\}} STDScore_i - \min_{i \in \{1, \dots, n\}} STDScore_i} \quad (3.13)$$

Where:

$$AvScore_i = \frac{PScore_i + NScore_i}{2} \quad (3.14)$$

$$STDScore_i = \sqrt{\frac{1}{2}((PScore_i - AvScore_i)^2 + (NScore_i - AvScore_i)^2)} \quad (3.15)$$

$$MaxScore = \max(\max_{i \in \{1, \dots, n\}} PScore_i, \max_{i \in \{1, \dots, n\}} NScore_i) \quad (3.16)$$

$$MinScore = \min(\min_{i \in \{1, \dots, n\}} PScore_i, \min_{i \in \{1, \dots, n\}} NScore_i) \quad (3.17)$$

A bar chart must be generated showing all the *aspects* of *entity e*. For each *aspect*, the size of the bar above the x-axis will correspond to its *positive score* and the size of the bar under the x-axis will be its *negative score*. Aspects could be ordered in the chart according to three different rules:

- According to *Relative Importance*
- According to *Positive Score*
- According to *Negative Score*

### 3.4.2 Using Regressions

Empirical findings support the notion that on-line consumer reviews can have a strong influence on the decision-making processes of potential buyers, who search the Internet for product information [95]. In fact, Hu, Liu and Zhang state that on-line product reviews provided by consumers who previously purchased products have become a major information source for consumers and marketers regarding product quality [96]. Regarding that, some publications have considered the structure generated by, for instance, opinion tuples and some similar approaches, to generate data as input for regressions that aim to directly estimate consumer preferences and their impact on product revenues.

One of the first works on this topic appears on the [50], where Archak, Ghose and Ipeirotis use techniques that decompose reviews into segments that evaluate the individual characteristics of a product and adapt methods from the econometrics literature (specifically the hedonic regression), to estimate the weight that customers place on each individual feature, the implicit evaluation score that customers assign to each feature, and finally, how these evaluations affect the revenue for a given product. Their results show that, by using product demand as the objective function for the regression, it is possible to derive a context-aware interpretation of opinions, also showing how customers interpret the posted comments and how they affect their choices. They proved the value of using an economic model to obtain

quantitative interpretations of consumer reviews on the web, since their results can be used by firms to determine which features contribute most to the demand for their product. Such information can also help manufacturers facilitate changes in product design over the course of a product’s life cycle as well as help to decide on which features to promote.

On the other hand, a more recent development by Decker and Trusov can be found in [51]. They presented an econometric model that can be applied to turn a set of individual consumer opinions, available as on-line product reviews, into aggregate consumer preference data. In their proposal, one major challenge is to accurately identify product attributes that appear in the reviews. In order to achieve this, the following steps were applied:

1. Review-wise partitioning of the pros and cons into individual words and phrases.
2. Elimination of those words and phrases that neither point to explicit nor implicit product attributes.
3. Aggregation of redundant words and phrases.
4. Transformation of implicit candidate attributes.
5. Merging of synonyms
6. Elimination of those candidate attributes that are less frequent.
7. Binary coding of the pro/con summaries using the available set of attributes.

Finally, they generate the following structure:

Review	Product	Attribute					Overall rating
		$l=1$	$l=2$	...	$l=L-1$	$l=L$	
$k=1$	E	pro	mv	...	con	mv	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$
$k=K$	C	con	pro	...	mv	mv	5

Figure 3.1: Preview of the product attributes identification process developed by Decker and Trusov.  
Source: [51]

Similarities with the aspect extraction process presented here are evident. Taking this as an input, they propose that each product review (consumer opinion) can be represented by a combination of positively (the pros) and negatively (the cons) valued characteristics, completed by an overall evaluation of the product. In consideration of this, they developed an econometric framework that includes a homogeneous preferences model, a heterogeneous model with a discrete distribution of preferences (the latent class Poisson regression approach) and a heterogeneous model with a continuous distribution of preferences (the negative binomial regression or NBR approach). In each model, parameters allow for inferences on the relative effect of functional attributes and brand names on product evaluations and purchasing decisions. Their results showed that the methodology they suggested is useful in collecting information about consumer preferences. As before, they also declare that results could be used in on-line shop optimizations, presumably affecting purchasing decisions.

As seen, using data about opinions to derive consumer preferences has proven to be a very promising task and many applications can be derived. For instance an interesting issue might be the comparison of consumer preferences elicited from on-line product reviews written by consumers of different countries. Empirical findings of this type might help to manage the balancing act between product individualization according to national preference patterns and standardization in favor of measure efficiency in international marketing.

On the other hand, reviewer profiling is another interesting topic. Moe and Schweidel [97] demonstrated that product opinions can be affected substantially by the composition of the considered consumer base, and identify four different types of review posters.

By adding a temporal dimension, it would be interesting to investigate how managerial actions such as promotions, price changes, new product introductions or competition influence consumer preferences, how they are reflected in product ratings and, in particular, the perception of attribute importance.

A somewhat different but related topic is discussed in [32], where Zhu and Zhang examine how product and consumer characteristics moderate the influence of on-line consumer reviews on product sales using data from the video game industry. Their results show a differential impact of consumer reviews across products in the same product category and suggest that firms' on-line marketing strategies should be contingent on product and consumer characteristics.

In conclusion, all the presented uses of opinions reinforce the development of techniques that are capable of generating a structure from them, encouraging studies like the one that is presented here.

# Chapter 4

## Design

In this chapter, all the models and methods discussed before will be used to design an application, which will put into practice given definitions and assumptions. This application will be described in general terms, while also defining its components and relations between them, without specifying implementation details.

### 4.1 General Requirements

In previous chapters, a set of models and tools have been developed in order to find preferences about products using opinions. In consequence, the application here designed will implement the aspect-based opinion extraction process that was introduced in the last chapter, use the output data of the model to find preferences about tourism products and show this information to users in a simple and intuitive manner. By implementing all the designed techniques in this application, it will be possible to put into practice all the models and theories proposed, proving if all these ideas are really feasible to be conducted and if they really permit the achievement of the objectives of this work.

To accomplish this, two different main tasks need to be performed:

- (a) In the first place, a set of definitions have to be given in order to make clear how the evaluation data for the developed aspect-based opinion mining algorithms will be structured (e.g. indicating the data source, size and other characteristics,) also specifying under what parameters these algorithms are going to be evaluated and how these parameters will be defined.
- (b) Secondly, the specifications or requirements of the web opinion mining application need to be stated, ensuring that the system includes the following functionalities:
  - (1) Collect opinions about Los Lagos tourism products from a given set of web sources. Data obtained from these sources will be input for the opinion extraction process and for the evaluation data in (a).
  - (2) Apply the aspect-based opinion extraction model, using data obtained in (1).

- (3) Provide users a platform to visualize and explore the mining results from (2), preparing aspect-based summaries from the opinions retrieved.
- (4) Evaluate the performance of the developed aspect-based opinion mining algorithms, according to definitions given in (a).

This chapter is structured as follows. In section 4.2, characteristics of the evaluation data are discussed and corpus specifications are given. Then, in section 4.3, evaluation parameters of the algorithms are defined and explained. Finally, in section 4.4, an opinion mining application is designed, considering all the specified requirements.

## 4.2 Annotated Corpus Structure

Requirements state that a method of measuring the performance of the opinion mining algorithms is needed. The first step then is, bearing in mind how each one of the algorithms work, to create a set of opinionated documents with all the known pairs  $(ae, s)$ . Using linguistic notation, each one of these special opinionated documents will be a linguistic corpus.

### 4.2.1 Corpus Description

When elaborating corpora, the most basic properties that need to be defined are language (or languages for multilingual corpora) and the nature of the text that the documents will contain. In this case, according to what this study proposes, corpora will be purely in English and will consider a set of reviews or comments regarding two different tourism products, hotels and restaurants. These products were chosen because they are part of the most common tourism activities, which also implies that reviews regarding them are frequently available on tourism web pages.

Corpora will consist of two documents, one containing data only about hotels and the other considering only data about restaurants. Also, since opinion extraction algorithms analyze text at sentence level, each corpus will be presented as a set of sentences for each review included. Documents will be disposed in plain text format (a .txt file), where each line will correspond to a sentence of a review, marked with the review number and the sentence number in order to make navigation across the file easier. Also, since algorithms are not case sensitive, all the words will be in lower-case.

One last important fact is related to the size that these corpora should have. In order to define this, a little glance at Bing Liu's work is presented. Since he also generated a specially annotated set of corpora to evaluate the performance of his algorithms (the same that inspired some of this work,) these corpora share a lot of similarities with what it is intended to be developed here. Liu's annotated corpora were elaborated for [65] and [64] during 2004 and 2008, respectively. They are all available on Liu's [web site](#).

The evaluation corpora are segmented into 8 documents, each document containing data on a different product. Each product has a set of approximately 100 reviews, obtained from the websites Amazon.com and C|Net.com. Information about each review considers its title,

the content of the review and some other data that was not used (the title was not used either). Below, a list of all the products is given:

- 2 digital cameras (2004)
- 1 DVD player (2004)
- 1 mp3 player (2004)
- 2 mobile phones (the second one added in 2008)
- A router (added in 2008)
- An anti-virus software (added in 2008)

From this list, only 2004 corpora were studied in detail. This was decided because information regarding how these corpora were constructed was more complete for them. Table 4.1 shows some interesting specifications of the mentioned corpora:

Product Name	Number of Reviews	Number of Sentences
Canon G3	45	597
Nikon Coolpix 4300	34	346
Nokia 6610	41	546
Creative Labs Nomad Jukebox Zen Xtra 40GB	95	1.716
Apex AD2600 Progressive-scan DVD player	99	739
Average	63	789
Total	314	3.944

Table 4.1: Details about corpora used by Liu and Hu to evaluate the performance of their algorithms. Source: Own elaboration using [65].

Here, average values of the number of reviews and the number of sentences in each review will be considered as good corpora specifications, as they somewhat represent the major tendency in existing related corpora. Nevertheless, without loss of generality and following the spirit of what Liu proposes in [65], it will be considered that a maximum of 100 reviews will be fairly sufficient. Considering this number of reviews, the number of sentences in each corpus should be a little over Liu’s average (789.)

Finally, in order to get reviews separately about hotels and restaurants, a web crawler program needs to be built. This program will be part of the application that will be designed later. Details about its construction will be given in the corresponding section.

## 4.2.2 Annotation Structure

In Liu’s work, each review was manually annotated. For each sentence in a review, if it showed any user’s opinion, all *aspects* that opinions were given on were tagged. Whether the opinion is positive or negative (i.e., the opinion orientation) is also identified. If the user gave no opinion in a sentence, it was not tagged. The following notation was used:



[t]:	the title of the review: Each [t] tag starts a review.
xxxx[+ -n]:	xxxx is a product <i>aspect</i> or <i>feature</i> .
[+n]:	Positive opinion, n is the opinion strength: 3 strongest and 1 weakest (strength is quite subjective).
[-n]:	Negative opinion.
## :	Start of each sentence. Each line is a sentence.
[u] :	<i>Aspect</i> or <i>feature</i> did not appear in the sentence.
[p] :	<i>Aspect</i> or <i>feature</i> did not appear in the sentence. Pronoun resolution is needed.
[s] :	Suggestion or recommendation.
[cc]:	Comparison with a competing product from a different brand.
[cs]:	Comparison with a competing product from the same brand.

Table 4.2: Bing Liu’s tagging structure for an evaluation corpus.  
Source: Evaluation corpora for [65].

Taking this as a basis, the proposed structure of the corpus for hotels and restaurants is shown in table 4.3, below.

\n :	Regular expression (but invisible character) that indicates the end of a line. Each line will contain only one sentence from one review.
[yyyy]:	Indicates that the line belongs to the comment number yyyy.
[zzzz]:	Indicates that the line is the sentence number zzzz for the current comment, indicated in [yyyy].
xxxx[+ -][u p]:	xxxx indicates that the <i>aspect</i> xxxx of the product (entity) appears in the sentence. If more than one <i>aspect</i> appears in the sentence, each <i>aspect</i> will be comma separated.
[+]:	Indicates if the sentence has a positive orientation on the <i>aspect</i> xxxx.
[-]:	Indicates negative orientation.
[u] :	Marks an <i>aspect</i> that appeared implicitly in the sentence.
[p] :	Marks an <i>aspect</i> that was indicated by a pronoun, i.e., requires pronoun resolution.
### :	Indicates the beginning of the original sentence.

Table 4.3: Corpus annotation structure proposed in this work.  
Source: Own preparation.

Clearly, there are some important differences between this proposal and the work used as inspiration. In the first place, as discussed in the previous section, indications about the sentence number and review number were added to facilitate corpus analysis and discussion when annotating or using its content. Also, any useless original information was omitted, avoiding overstating and increasing the corpus readability. Finally, the proposed notation does not consider comparative opinions as special cases.

### 4.2.3 Tagging Methodology

According to Hu and Liu in [65], the annotating process was quite straightforward for both *aspects* and opinions. Nevertheless, a minor issue regarding *aspect* tagging was found; they could appear explicitly or implicitly in a sentence (as seen, annotation structure covers these cases). Also, they mention that judging opinions in reviews can be somewhat subjective. However, they declare that it was usually easy to judge whether an opinion was positive or negative if a sentence clearly expresses an opinion. For all difficult cases, a consensus was reached between the primary human tagger (the first author of the paper) and the secondary

tagger (the second author of the paper.) In [64], the same problems appeared and the proposed solution was the same, reaching a consensus between the primary human taggers (the paper author and two master's degree students) and the secondary tagger (the second author of the paper.)

In contrast, inside evaluation corpora files, Hu and Liu indicate that tagging is a difficult task and that errors and inconsistencies are inevitable. Here, it has also been stated that finding opinions in opinionated text could be a very subjective task, mainly because options that language provides to give opinions are practically infinite. Bearing that in mind, a very simple and compact set of tagging rules has been developed, in order to help and guide the tagging process in most of the possible cases. These rules have been developed after a very exhaustive review of different corpora (including the ones that were generated in this work) and trying to generalize as many cases as possible. Nevertheless, there will certainly always be a lot of uncovered cases in practice, which must be analyzed considering the spirit of the tagging process. The list of rules is presented below.

### ***Aspect Rules***

Base Rule: An *aspect* appears in a sentence if any word in it implies its presence directly or indirectly.

**Rule 1.1** An *aspect* appears explicitly in a sentence if there is a noun or noun phrase that indicates it directly. Every time an *aspect* appears explicitly in a sentence and an opinion is given on it (see Rule 2), it must be tagged using the normalized (i.e. singular) *aspect expression* with the sign of the corresponding opinion orientation. Also, the most general *aspect expression* should be chosen (avoid tagging different *categories* of an *aspect*).

**Rule 1.2** An aspect appears in an implicit way if, in a sentence, there is not any noun or noun phrase that indicates it directly, but its presence can be deduced from the context. Some common implicit *aspect expression* indicators are adjectives, pronouns and also some verbs. If an *aspect expression* appears in an implicit way and an opinion is given on it, it must be tagged using the [u] notation.

**Rule 1.2.1** A special circumstance occurs when the *aspect expression* indicator is a pronoun, in which case, the [p] notation must be used instead of [u]. This is used to indicate that pronoun resolution is needed.

**Rule 1.2.2** In the case that a proper noun is used to implicitly indicate an *aspect expression*, the underlying general *aspect expression* that is implied or the closest known aspect *should* be tagged. A proper noun must never be an *aspect expression*.

**Rule 1.3** If an opinion is given on the *entity* (the concept that defines the corpus) it must be tagged as if it was an *aspect expression*.

### **Opinion Rules**

Base Rule: It will be considered that the user gives an opinion on an *aspect* if in the sentence where the *aspect expression* appears, a positive or negative feeling toward it is transmitted.

**Rule 2.1** The most common case is when the feeling is mostly expressed or indicated by special sentiment words, word groups or comparative references. In this case, the opinion orientation is direct and given by the orientation of the nearby words.

**Rule 2.2** Recommendations, thanks or acknowledgements and congratulations are considered positive opinions. *Aspect expressions* appearing in these sentences should be tagged with a positive orientation.

**Rule 2.3** When a sentence declares the absence of any *aspect* (for instance using the words *have no* or any other equivalent clause,) the corresponding *aspect expression* must be tagged with a negative orientation. This is proposed since in this case, the user is declaring that the *entity* did not have a component or attribute (i.e. *aspect*) that it should have had. Conversely, when a sentence declares the presence of any *aspect*, the corresponding *aspect expression* should be tagged with a positive orientation.

### Completeness Rules

**Rule 3** If an opinion is given, but it is not possible to clearly identify the opinion target (i.e. the *aspect*), it will be assumed that the entire *entity* is the target of the opinion.

**Rule 4** If an explicit *aspect expression* appears and no opinion is given on it, but an opinion is given on a second *aspect expression* (that could explicitly or implicitly appear) and this *aspect* is a subcategory of the first one, both of them must be tagged with the same orientation given by the opinion in the sentence. This situation appears commonly when in a sentence an *aspect* appears and the same sentence also indicates the absence of any attribute or subcomponent that is an *aspect* itself (see Rule 2.3).

### Tagging Examples

In this section, a complete set of tagging examples, based on the rules already proposed, is presented. The objective of including these examples is to show how rules should be applied when tagging. In the following cases, comment and sentence numbers are fictitious, and the following notation is used:

- Let  $s$  be a sentence about entity  $e$ .
- $A$  will be an arbitrary *aspect* that explicitly appears in  $s$ . If  $A$  appears implicitly in  $s$ , it will be denoted as  $A'$ .
- We will denote  $a$  to any *aspect* that is a subcategory of  $A$  (or  $A'$ ) that appears explicitly. Likewise, if  $a$  appears implicitly, it will be denoted as  $a'$ .

#### 1. Application of rule 2.3 and 1.1

[c22][s5] room[-], hot water[-], staff[-] ### my room<sub>A</sub> had no hot water<sub>a</sub> and the staff<sub>B</sub> was unconcerned and unhelpful.

Where:  $hotel(e) \rightarrow room(A) \rightarrow hot\ water(a)$ .

2. Application of rule 2.3 and 4

[c22][s4] room[-], internet[-] ### my room had no internet, either.  
A a

Where:  $hotel(e) \rightarrow room(A) \rightarrow internet(a)$ .

3. Application of rule 1.2

[c26][s14] staff[+] ### thank you to all the staff, and particularly christine.  
A A

Where:  $hotel(e) \rightarrow staff(A) \rightarrow christine(\text{person name for } A)$ .

4. Application of rule 1.2, 1.1 and 2.1

[c2][s1] hotel[+], acommodation[+], location[-][u] ### the actual hotel  
accomodations were very luxurious and absolutely wonderful, but antumalal  
B  
was too far away from the actual town of pucon.  
A'

5. Application of rule 1.1. and 2.1

coffee[-] ### we waited for 30 minutes and then the waiter brought us a very bad  
coffee.  
a

Where:  $restaurant(e) \rightarrow drink(A) \rightarrow coffee(a)$ .

6. Application of rule 2.3 and 3

[c11][s9] hotel[-], dinner[-] ### no fancy dinners here.  
indicates absence indicates entity

7. Using rule 2.2

[c45][s12] hotel[+] ### this is quite the most delightful hotel i have stayed in for a  
very long time.  
e

Where:  $hotel(e) \rightarrow hotel(A)$

8. Application of rule 1.3 and 2.2

[c55][s5] hotel[+] ### i would definitely recommend the hotel.  
recommendation e

Where:  $hotel(e) \rightarrow hotel(A)$

Finally, the following table (4.4) graphically shows how a tagged set of sentences should look:

line	
1	<i>[c1][s1]</i> place[+], comfort[+][u], location[+][u] ## a good place to stay at the end of a long flight in that it is very comfortable, with many facilities, and in the town , by the shore .
2	<i>[c1][s2]</i> ### however, puerto montt is not the best of places to explore.
3	<i>[c1][s3]</i> ### a better place is puerto varas which is just as near to the airport and has far more attractions.
4	<i>[c1][s4]</i> ### could not fault this aparthotel.
5	<i>[c2][s1]</i> hotel[+] ### my fiance and i spent three nights here in march 2012 and it's a sweet, quaint hotel.
6	<i>[c2][s2]</i> reservation[-] ### with that said, i called in early february 2012 to make a reservation and it got lost/misplaced.
7	<i>[c2][s3]</i> staff[+] ### joyce and jose luis really treated us great, i would not hesitate to stay there again.
8	<i>[c3][s1]</i> hotel[+], attention[+], staff[+], food[+], location[+], room[+] ### excellent hotel, attention to the details, great staff and food, amazing room, location is just excellent and quiet.

Table 4.4: Tagged sentences example.  
Source: Own elaboration.

### 4.3 Performance Evaluation Parameters

To evaluate the performance of the developed opinion mining algorithms, three main tasks have been defined. These tasks are intrinsically related with the processes of finding aspect expressions and determining orientation (as seen in Chapter 3) and also consider some other perspectives found in literature. Basically, they correspond to classical classification problems in Machine Learning, hence the following standard performance measures will be obtained in each case:

- Accuracy
- Precision
- Recall
- F-measure

Although these performance measures are common to all classification problems, each task has some special conditions that somewhat change certain details about how these measures must be obtained. Also, since several assumptions are usually imposed in NLP in order to achieve a better understanding of a problem, some definitions change depending on what assumptions are considered.

Bearing that in mind, the three main tasks are presented below. In each case, a short explanation that helps in understanding its objective is presented, also including different possible definitions and ways that the classical performance measures will be calculated.

### 4.3.1 Aspect Extraction

Aspect extraction is the first task that will be evaluated. In general terms, this task corresponds to the process of finding the *aspects* that appear in a set of opinionated documents of a given set of *entities*. Nevertheless, the problem here is a little different. In the first place, the *entity* in each opinionated document will simply depend on the corpus, being only hotels or restaurants. Secondly, the developed algorithm is only capable of finding the *aspect expressions* (i.e. only aspects that are nouns or noun phrases.) Thus, in this case the classification problem is simply declaring whether a set of stems is or is not an *aspect expression*. Taking into consideration these special conditions, let  $AC_i = \{ac_{i1}, ac_{i2}, \dots, ac_{in}\}$  be the set of *aspect expressions* annotated in the corpus of the *entity*  $e_i$  and let  $AA_i = \{aa_{i1}, aa_{i2}, \dots, aa_{im}\}$  be the set of *aspect expressions* found by the algorithm. Then:

- True Positives: Is the set of *aspect expressions* that appears in both sets,  $TP = AC_i \cap AA_i$ . In other words, these *aspect expressions* have been detected by the algorithm and were annotated in the corpus by the human tagger.
- False Positives: Correspond to the set of *aspect expressions* that have been extracted by the algorithm but were not annotated by the human tagger,  $FP = AA_i \setminus AC_i$ . In simple terms, these *aspect expressions* were wrongly extracted by the algorithm.
- False Negatives: Is the set of *aspect expressions* that were not extracted by the algorithm but were annotated by the human tagger,  $FN = AC_i \setminus AA_i$ . Therefore, this is the set of *aspect expressions* that should have been extracted by the algorithm, but were not.
- True Negatives: Theoretically, corresponds to the set of items that were classified as *not aspects* by both the human tagger and the algorithm. In this case, this set is empty by definition; a set of stems is not extracted from the text if it is not considered an *aspect expression*.

Considering these definitions, two possible definitions for the task of extracting *aspects* are proposed:

- Under a first proposal, it will be considered that the complete set of *aspect expressions* that appear in the corpus is the final objective of the algorithm. This assumption makes sense supposing that the algorithm is capable of extracting both explicit and implicit *aspect expressions*, something that as stated before, is not possible. Nevertheless, although this maximum performance is unreachable for the algorithm, it seems reasonable to wonder how it performs under real conditions, i.e., how it performs in the task of extracting all the *aspect expressions*. This special task will be called **Total Aspect Extraction**.
- Under a second perspective, the final objective of the algorithm will be the set of all explicit *aspect expressions* appearing in the corpus. Under this condition, the best performance of the algorithm will be its maximum feasible set of extracted *aspect expressions*. This task will be henceforth called **Explicit Aspect Extraction**.

### 4.3.2 Sentence Subjectivity Classification

The second task is sentence subjectivity classification, i.e., the process of detecting those sentences that contain opinions. In literature, this task is called Subjective Classification by most authors, but some of them specify certain special details regarding this task, indicating that a subjective sentence will not necessarily contain an opinion and, conversely, that an opinion will not necessarily imply a subjective sentence. While here these concepts are recognized as different, this work will consider two assumptions in defining a subjective sentence. Let  $oc$  be a sentence in one corpus,  $oc \in OC$ , then:

- A first possible definition is declaring a sentence as subjective simply when any *aspect* is mentioned in it. Although this definition seems valid but somewhat far from intuitive, its importance lies in the fact that this perspective allows measuring how effective the *aspect* extraction algorithm is in finding all those sentences that contain any *aspect*. The task of sentence subjective classification under this definition will be called **Subjectivity Classification**.
- On the other hand, a second approach will consider a sentence as subjective when an aspect appears in it and sentiment orientation on that aspect is different from 0. In other words, a sentence will be subjective when an *opinion word* is used to describe any appearing *aspect*. In this work, this special task will be called **Subjectivity Classification (without 0)**.

Let  $OSC = \{osc_1, osc_2, \dots, osc_n\}$  be the set of subjective sentences found in one corpus by the human tagger. Also,  $OSA = \{osa_1, osa_2, osa_3, \dots, osa_m\}$  will be the set of subjective sentences detected by the algorithm and let  $S$  be the set of all the sentences in the corpus.

- True Positives: Is the set of sentences that belong to  $OSC$  and  $OSA$ ,  $TP = OSC \cap OSA$ . These sentences have been classified as subjective by the algorithm and the human tagger.
- False Positives: Is the set of sentences that have been classified as subjective by the algorithm, but not by the human tagger,  $FP = OSA \setminus OSC$ .
- False Negatives: Corresponds to the set of subjective sentences that has been detected by the algorithm, but were not annotated in the corpus,  $FN = OSC \setminus OSA$ .
- True Negatives: Is the set of sentences that have not been classified as subjective by either the algorithm or the human tagger,  $TN = (S \setminus OSC) \cap (S \setminus OSA)$ . In other words, these sentences were correctly classified as not subjective by the algorithm.

### 4.3.3 Aspect Sentiment Classification

The last task that will be evaluated is aspect sentiment classification, or in other words, the process of determining if a tuple  $(ae, s)$   $ae \in AE, s \in S$  is positive, negative or neutral. It is important to mention that in this case, since not all the existing *aspect expressions* will be extracted by the algorithm, the performance evaluation of this task will be limited to the set of tuples detected by both the algorithm and the human tagger. Also, since the

neutral category might be indicating that no sentiment orientation is given in the sentence, it seems reasonable to consider those cases specially. So, given that omitting the *neutral* class the classification problem turns binary (there are only 2 classes,) only the performance of the *positive* class will be evaluated, since the performance of the other is complementary. Considering all mentioned specifications, two different perspectives are proposed:

- A first approach considers that since the human tagger does not classify sentences as neutral (if a sentence has no opinion on an *aspect* it is simply not annotated), tuples classified in this category will correspond to an algorithm error. In other words, the *not positive* class will consider both negative and neutral tuples. This task will be simply called **Sentiment Classification**.
- On the other hand, a second perspective will ignore tuples classified as neutral, assuming that they were not subjective. This special task will receive the name of **Sentiment Classification (without 0)**.

Then, let  $TC = \{tc_1, tc_2, \dots, tc_n\}$  be the set of the positive classified tuples in the corpus and let  $TA = \{ta_1, ta_2, \dots, ta_m\}$  be the set of positive tuples extracted by the algorithm.

- True Positives: Corresponds to the set of tuples that were classified as positive by the algorithm and the human tagger,  $TP = TC \cap TA$ .
- False Positives: Is the set of tuples that were classified as positive by the algorithm but not by the human tagger in the corpus.
- False Negatives: The set of tuples that were classified as positive in the corpus but not by the algorithm. In other words, these tuples should have been classified as positive by the algorithm, but they were not.
- True Negative: Is the set of tuples that were not classified as positive by both the algorithm and the human tagger.

## 4.4 Application Design

Considering the general requirements specified, modular programming will be used to implement the application. Under this paradigm, design emphasizes separating the functionality of a program into independent, interchangeable modules. The main idea is that each module must contain everything necessary to execute one aspect of the desired functionality.

A glance at the modular programming philosophy can be found in a 1970 textbook on the design of system programs, by Gouthier and Pont [98], declaring that a well-defined segmentation of the project effort ensures system modularity. Each task forms a separate, distinct program module. At implementation time, each module and its inputs and outputs are well-defined and there is no confusion in the intended interface with other system modules. In other words, with modular programming, concerns are then separated such that modules perform logically discrete functions. Each module (which can contain a number of separate processes) works independently from another module. In this way, modularization



improves the flexibility and comprehensibility of a system while allowing the shortening of its development time [99]. In this case, since functionalities have been declared through the general requirements, each one representing a different function, modularization functions as a great, if not the best, programming paradigm.

The general architecture of the application will be is composed of five modules, called Data Persistence Module (DPM), Data Collecting Module (DCM), Opinion Mining Module (OMM), Results Visualization Module (RVM) and Performance Evaluation Module (PEM). Figure 4.1 shows how these five modules interact:

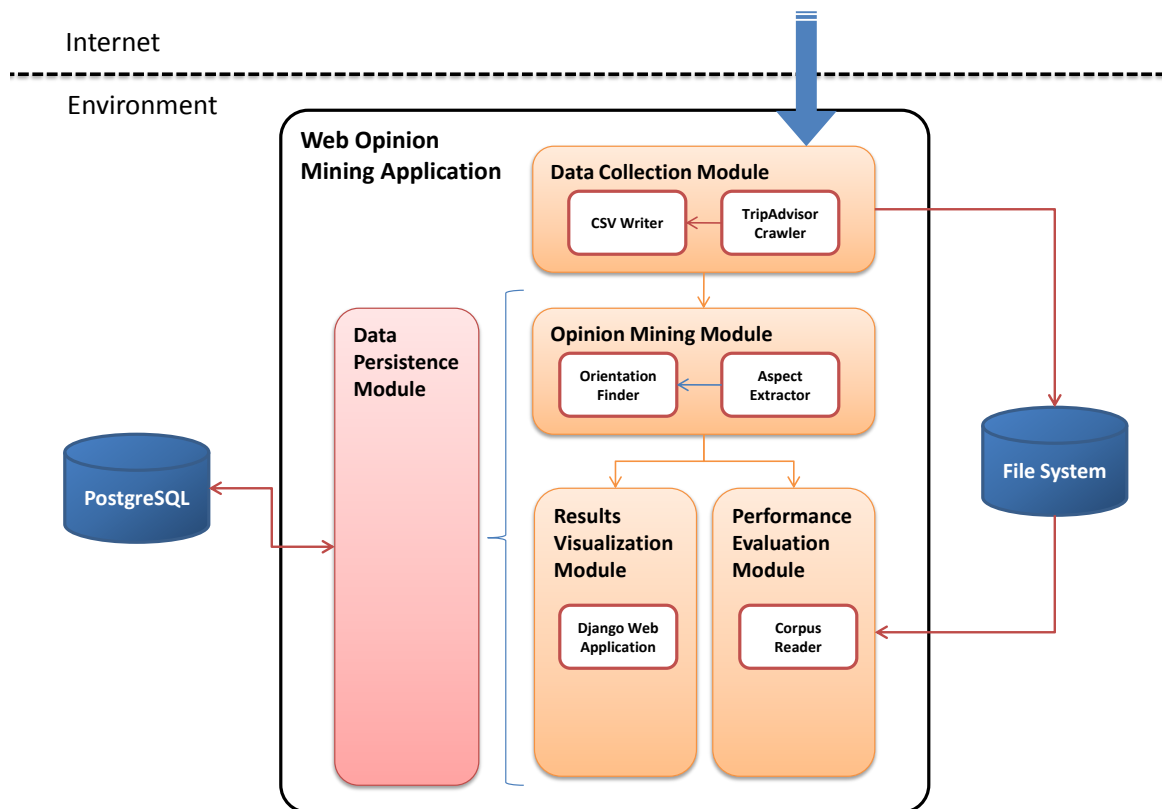


Figure 4.1: General design of the Web Opinion Mining Application.  
Source: Own elaboration.

In the following sections, the functionalities of each module are detailed, including use cases and other diagrams that might be necessary.

#### 4.4.1 Data Collection Module

The Data Collection Module (DCM) will be in charge of obtaining opinions from a set of given web sources. These opinions will be the source for the application and will also be used in constructing the annotated corpus for evaluation. So, the first step is defining what sources will be considered.

Based on the work of [100], which studies tourism-oriented web sites, analyzing their structure, functionalities and reporting important facts and statistics about them, the website

**TripAdvisor** has been chosen as the main data source. TripAdvisor is one of the biggest travel websites around the world, offering to visitors some tools to plan better trips. In particular, the site offers advice from real travelers to the user (a potential traveler,) about a wide variety of destinations, also adding the possibility of connecting to other websites or reservations systems for hotels, restaurants and other similar products or services. This advice is presented in the way of reviews or opinions that only registered users can post. Every time a review is published by any user, TripAdvisor's staff checks its content. In case the content is considered inadequate, the post is eliminated, generally without notifying the user. According to experience, this validation process often occurs one or two days after the review is published.

In addition, each review often includes some trip recommendations that travelers give to users who are planning their own travel. Besides opinions, reviews or advice include a user rating, which is given using a 0 to 5 scale (6 levels.) There is also a chance for the service or product provider to reply to the review with a short answer. Finally, for each review, users can indicate if it was helpful.

As stated in the official website, TripAdvisor sites make up the largest travel community in the world, with more than 60 million unique monthly visitors and over 75 million reviews and opinions. The site currently operates in 30 countries, but, since this work proposes a study on a specific geographic zone, it is only needed to obtain opinions about the X Región de Los Lagos, involving one country, Chile. The 30 countries included in TripAdvisor sites do not have their own exclusive platforms, but rather, each one has a different web domain with the proper region or zone orientation, according to the geographic location and language. For each domain, reviews are automatically translated to the proper language when needed.

In the absence of an exclusive platform for Chile, to access any tourism information in Spanish about this country, the site <http://www.tripadvisor.com.ar/Tourism-g294291-Chile-Vacations.html> has to be visited, while in order to get the same information in English, users should go to <http://www.tripadvisor.com/Tourism-g294291-Chile-Vacations.html>. The former corresponds to the version for the U.S., and the last is the Argentine version of TripAdvisor. In fact, given the domain-based system where users are redirected to a specific TripAdvisor version, it is also possible to visit a version in one particular language through different domains. For instance, the site is available in Spanish using the .ar or .es domains.

Within the information on the web page for Chile (as for any other location or zone in any language,) four main tourism product categories are available:

- Hotels
- Restaurants
- Vacation Rentals
- Activities or things to do

Each one of these items appears for every different Chilean geographic zone. In TripAdvisor, in November 2012, these zones were the ones below:

- Tarapaca Region
- Arica and Parinacota Region
- Atacama Region
- Coquimbo Region
- Valparaiso Region
- Lake District
- Easter Island
- O'Higgins Region
- Biobio Region
- Patagonia
- Santiago Metropolitan Region
- Valle Central
- Aysen Region

As seen, region segmentation offered in TripAdvisor may or may not be correlative to Chilean regional distribution. Nevertheless, it is possible to visit the site for each available city or location in the X Región de Los Lagos, based on previous knowledge of which zones contain any of them. Despite that, in this work, the zone named the *Lake District* (in Spanish *Distrito del Lago*) will be considered as almost equivalent to the X Región de Los Lagos.

A comprehensive review showed that the *Lake District* includes all of the most important cities and localities in Los Lagos, although, it also considers some nearby cities belonging to other Regions. Frutillar, Puerto Octay, Pucón, Villarrica, Puerto Montt, Puerto Varas, Chiloé (including Castro and Ancud,) Temuco (part of the Región de la Araucanía) and Valdivia (the former Los Lagos portion, now in the Region de Los Ríos) belong to the *Lake District*. In this manner, the *Lake District* becomes the only source for this work, making the data obtention process a lot easier and introducing minimal noise or unwanted data.

Having defined the data source, the data collection process now needs to be specified. This task will be performed by a web crawler, which will download and pre-process data from TripAdvisor in relation to a *topic*. A *topic* is defined as a function of the following parameters:

- *Domain*: Only .com. This setting assures that only reviews originally written in English are retrieved, avoiding the automatic translation system that TripAdvisor implements.
- *Geographic Zone*, defined according to TripAdvisor's region segmentation. In this case, the *Lake District*.
- *Content* (specific product or service) of opinions:
  - Hotels
  - Restaurants

- *Language*: English, to avoid spelling mistakes and ensure a better text quality.

The crawler will work in three stages:

- (1) Obtaining and downloading all HTML pages in relation to a topic
- (2) Preprocessing downloaded pages in order to extract the following pieces:
  - Product Name
  - Address – City – Country - Zip Code
  - Review Title
  - Review Content
  - Name of the review’s author
  - Place where review’s author is from
  - Review Rating (from 0 to 5 stars)
- (3) Only comments originally written in English must be selected, detecting and deleting entries that might have been automatically translated. Once all this data is extracted, it is stored in a CSV file, which must be comma separated.

Finally, putting it all together, the following use case diagram summarizes all mentioned requirements in this section. See figure 4.2.

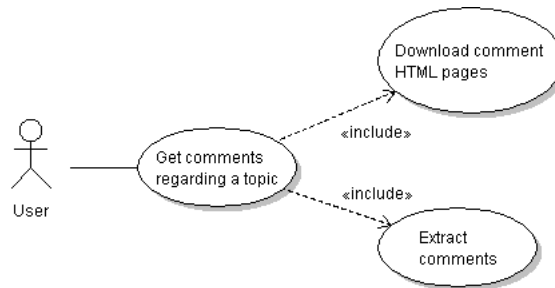


Figure 4.2: Data Collection Module use cases.  
Source: Own elaboration.

## 4.4.2 Results Visualization Module

The RVM is the main visible portion of the whole application that interacts directly with the user. Its main goal is to permit a user to extract opinions from a set of opinionated documents and explore the results of that process. These results include aspect-based opinion summaries and other related features, which then give insights to users about customer preferences in relation to tourism products in Los Lagos. As a result, requirements for this module will determine some specifications for the rest of them, which is why this section is presented first. Figure 4.3 shows the RVM use cases, which give a general concept of the services that will be available for the user. These services are explained below.

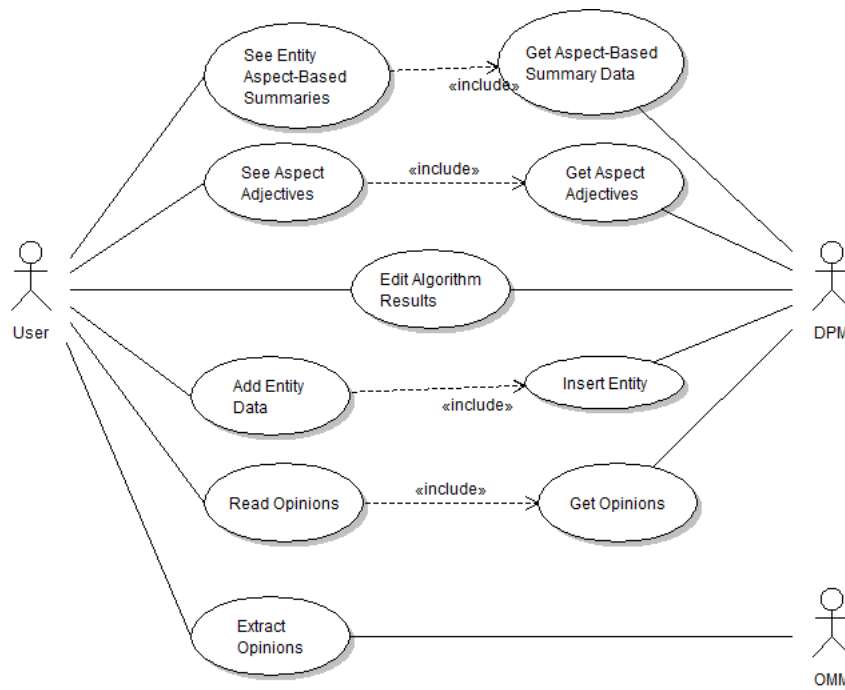


Figure 4.3: Results Visualization Module use cases.  
Source: Own elaboration.

1. Add Entity Data is the first activity that needs to be performed by the user (in order to access other services,) providing opinion data to the system. Data must be given to the system in the way of a CSV file containing opinions, comments or reviews regarding a *topic*, as defined in the previous section. In the system, data about a *topic* will appear in the manner of an *entity*, with the name denoting the added CSV file.
2. Extract Opinions: Once data has been added to the system, the corresponding *entity* will be available to apply the aspect-extraction process on its data. As a result, this process will give the user access to the following features:
  - See Aspect-Based Summaries, including bar charts, as proposed in Chapter 3, and a list of the extracted *aspects*, showing their *relative importance*. As discussed before, these summaries offer quantitative measures of consumer preferences on the *entity* (product) and of the relative importance of each attribute or component of the *entity*. The chart should look like figure 4.4.

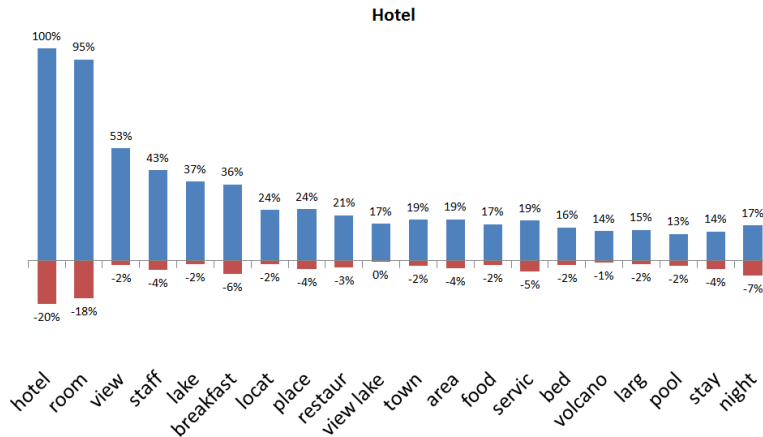


Figure 4.4: Aspect-Based summary bar chart example for the entity *hotel*. Source: Own preparation.

Source: Own elaboration using MS Excel.

- See Aspect Adjectives: Nearby adjectives in all sentences where an *aspect* appears will be shown in a bubble chart. This idea was inspired by the website [TinkTag](#), a site that let users leave their impressions about people that are frequently part of any news in Chile. Each user leaves his impression using only 20 characters. From this short sentence, important words are extracted and added to a special chart, like the one shown in the figure below. The final objective of this *game* (as the site calls itself) is to discover the Collective Impression of each added entity. Figure 4.5 presents an example of the charts appearing in TinkTag.



Figure 4.5: Screen shot of the TinkTag website, showing its special impression-based chart on Sebastián Piñera (entity).

Source: TinkTag web site.

In this case, adjective bubble charts about an *aspect*, are intended to offer a qualitative description about customer preferences on that *aspect*, complementing information displayed in aspect-based summaries. Bubble charts should look as

figure 4.6.

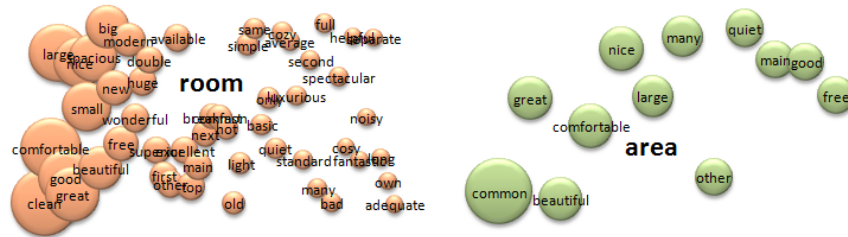


Figure 4.6: Adjective bubble chart examples for the *entity* hotel and its *aspects* room and area.  
Source: Own elaboration using MS Excel.

In the two last cases, considering that *aspects* will essentially be a set of stems, original words that were part of each *aspect* should be displayed to avoid ambiguities (caused by word stemming) and ensure that results are intuitive.

- Read Opinions: A list of all original sentences where an *aspect* appears will be displayed in an ad-hoc manner, classifying them as positive or negative. Original sentences are shown to give user access to the *real data*. This feature could also be included considering the whole *entity* and showing or marking on each sentence, those *aspects* that were detected.
- Alter Algorithm Results: Since the developed algorithms will probably not cover all possible cases, the possibility of exploring original sentences and editing algorithm results (*aspects* that were extracted and the orientation of each one) should also be offered to the user. This functionality will appear in a special menu that does not interfere with the specifications shown above, since it is not related to exploration. This feature is based on Opinion Observer, the opinion mining platform proposed by Bing Liu in [75], which offered users a tagging interface in order to improve the system performance. A screen shot, taken from the same paper, is attached below (see figure 4.7) to give an idea about how this service should look.

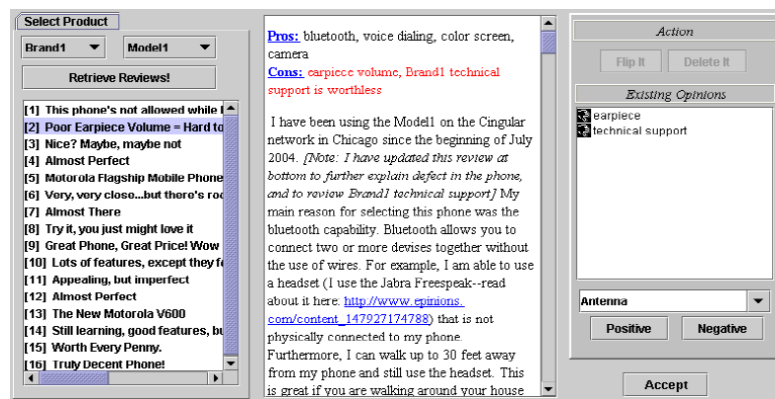


Figure 4.7: Tagging interface of Opinion Observer.

Source: [75]

### 4.4.3 Opinion Mining Module

The Opinion Mining Module implements the aspect-based opinion mining algorithms on a given set of opinionated documents, taking into consideration the opinion-extraction model developed in Chapter 3.

In the first place, the sentences of each review are split into tokens and POS tagging is applied. Then, two different tasks need to be performed, *aspect expression* extraction and orientation determination. Following the modular programming paradigm, two sub-modules are proposed:

#### Aspect Extraction Sub-Module

This sub-module will be in charge of applying the *aspect expression* extraction algorithm to a set of POS-tagged sentences. However, depending on the module that performs the request of extracting aspect, some special features need to be provided. For this, two different use cases are proposed.

##### 1. Simple Aspect Extraction

This task will be requested by the PEM and consists of a basic *aspect expression* extraction process that must give the following results:

- The set of extracted *aspect expressions* (as stem sets),  $AE$ , including the frequency of each one.
- The set of sentences that include any aspect,  $S$ . Each sentence must be related to the opinionated document to which it belongs.
- The set of extracted pairs  $(ae, s)$   $ae \in AE, s \in S$ , including the *position* of every *aspect word* in  $s$ . Orientation on each tuple will be set to 0.

##### 2. Aspect Extraction

This task will be requested by the RVM. To the results delivered by the Simple Aspect Extraction process, additional outputs are added:

- For each extracted *aspect expression*, the set of original words that originated its stems, also including their frequency as part of that *aspect expression*.
- Also, for each extracted *aspect expression*, nearby adjectives that appear in a sentence must be returned.

Both tasks include the task of Insert Aspect, which is requested to the DPM since it involves database operations. This module will be in charge of saving all the information delivered by this module in the best possible way.

#### Orientation Determination Sub-Module

This sub-module applies the orientation determination algorithm to the set pairs  $(ae, s)$  extracted by the sub-module above. Each pair will be assigned an orientation of 1 (if positive)



or -1 (if negative), according to the algorithm developed in the last chapter. If no orientation is detected (neutral), it will remain 0 as set before. Besides orientations of each pair  $(ae, s)$ , this sub-module will return the set of adjectives that appeared near each *aspect expression*, including their frequency in relation to that *aspect expression*. Then, the module takes the results and sends a request to the DPM, which updates the orientation of each tuple. Figure 4.8 shows the use cases of the OMM.

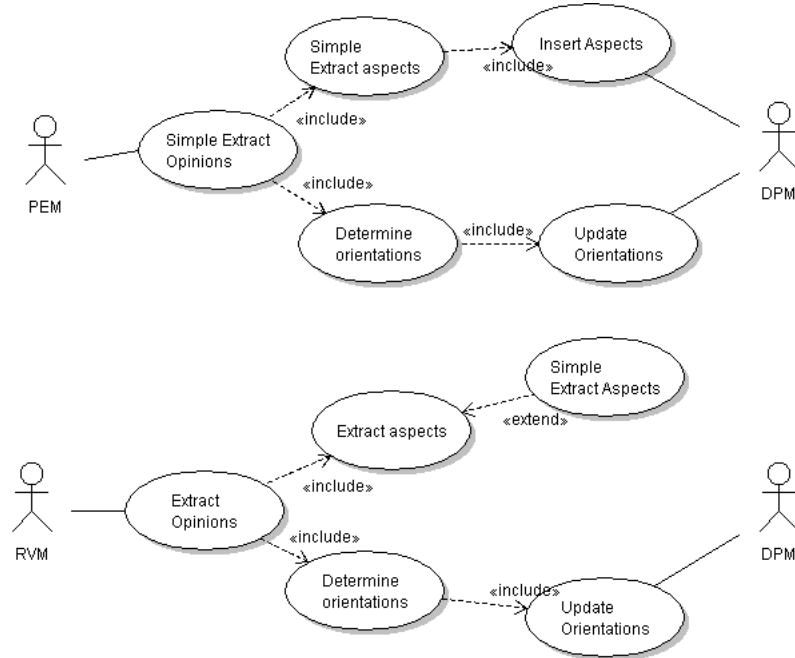


Figure 4.8: Opinion Mining Module use cases.

Source: Own elaboration.

#### 4.4.4 Performance Evaluation Module

The PEM is in charge of delivering a set of indexes that evaluate the performance of the opinion mining algorithms according to tasks already discussed. Figure 4.9 shows the use cases.

- The task Insert Human Tagged Data, consists of taking a .txt file containing an annotated corpus and saving its data (*entity*, sentences, *aspects* and orientations) in the database. In order to do this, the module reads the file, parses its content and requests a database operation from the DPM, which handles all insertions. Also, this process needs to add the corresponding human tagger model, also requesting this operation from the DPM.
- After setting the parameters of the model a user wishes to use, the task Apply Model will permit that user to insert data regarding that model's parameters in the database (through the DPM) and send a request to the OMM, which extracts and inserts (again, by calling the DPM) the aspects extracted from a previously selected corpus with their respective orientation.
- The Query Model Performance operation permits a user to see all the different evaluation measures for a model. The data required to calculate all the measures is requested

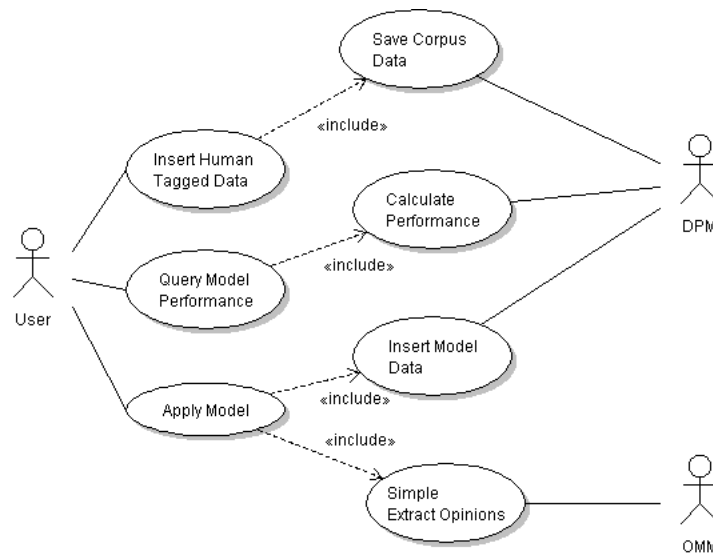


Figure 4.9: Performance Evaluation Module use cases.

Source: Own preparation.

from the DPM, while the data processing needed to obtain the evaluation measures is executed by the PEM.

#### 4.4.5 Data Persistence Module

The Data Persistence Module manages all database operations, constituting a model layer for the whole system. This module does not interact directly with the user, but with other modules that require any CRUD operation. Therefore, the DPM must support the database-related tasks of all other modules. Figure (4.10) delineates the use cases.

The rest of this section is structured as follows. In the first place, database E-R models will be presented, explaining how they support all the data that needs to be stored and all the operations that need to be performed. After that, details on functionalities will be given, considering the three different modules that interact with the DPM.

##### Database E-R Models

Considering that data structure must support all the different services that are provided by the system, two different E-R models will be developed. The first one will support data needed to visualize and explore the aspect-based summaries and related features, while the second one will deal with data that the PEM needs. Below, each model is explained.

- Application Model

Figure (4.11) shows the E-R model that will support all data that the RVM needs to perform all the requests made by the users. As can be seen, some of these requests also include interaction with other modules. The structure also considers data that these modules need to store.

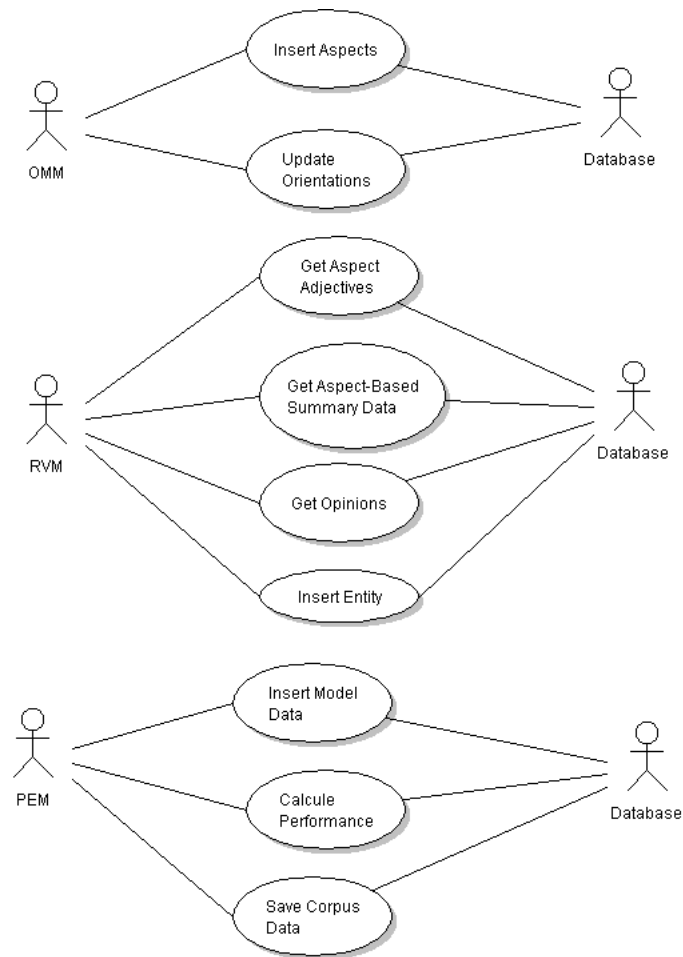


Figure 4.10: Data Persistence Module use cases.  
Source: Own elaboration.

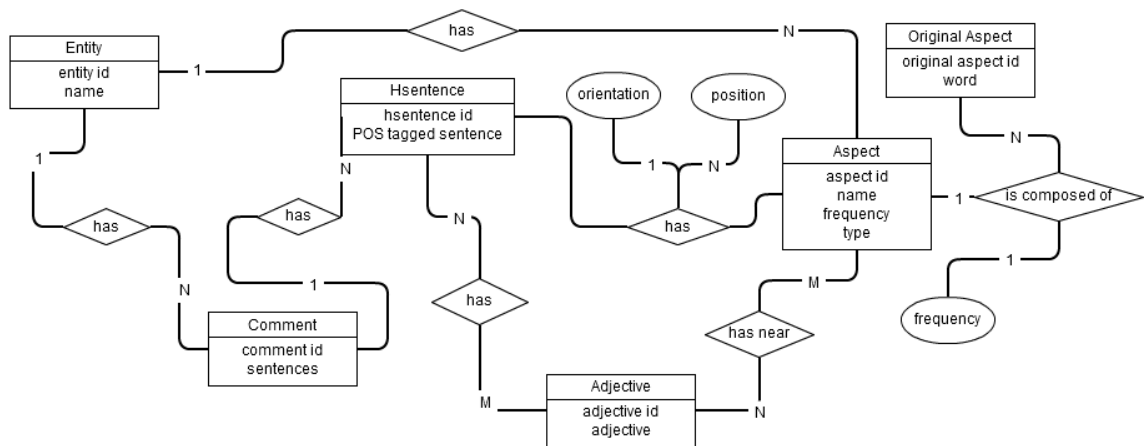


Figure 4.11: E-R model of the database for the application.  
Source: Own elaboration.

- **Entity**: Represents an *entity*, inserted using the corresponding CSV file.
  - **Aspect**: Represents an *aspect*, extracted from a set of opinionated documents or comments regarding an *entity*.
  - **Comment**: Represents a comment or review, as obtained from the CSV file. Sentences of the comment are saved as an attribute.
  - **Hsentence**: Is a sentence of a review in which the algorithms have extracted any *aspect*. The POS-tagged sentence is saved as an attribute.
  - **Adjective**: Represents an adjective appearing near an *aspect*.
  - **Original Aspect**: Represents a word that originated as any part of an *aspect expression*.
- Performance Evaluation Model

The following figure shows the developed E-R model that will make all database-related PEM operations possible. Since most of the data that this model needs to support is practically the same as before, the structure shares some of the features with the application model. Nevertheless, some objects were eliminated as unnecessary and also a special new entity was added. See figure 4.12 below.

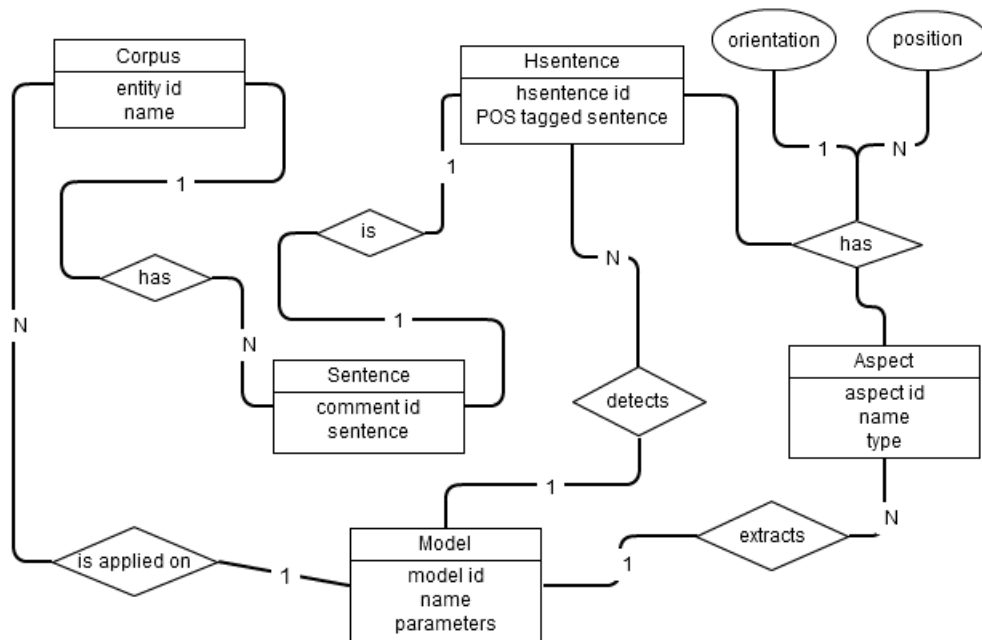


Figure 4.12: E-R model of the database for performance evaluation.  
Source: Own elaboration.

- **Entity**: Represents a *corpus* regarding data about an *entity*.
- **Aspect**: Represents an *aspect* extracted (by the human tagger or by the algorithms) from one *corpus*.

- **Sentence:** Represents a sentence from a *corpus*.
- **Model:** Represents a method of extracting *aspects* from a *corpus* in the database. In this manner, models will equally represent the human tagger or the algorithms.
- **Hsentence:** Is a sentence from a *corpus* in which a model has extracted any *aspect*. In case the algorithm extracted the sentence, the POS- tagged sentence is saved as an attribute.

## OMM Tasks

The Opinion Mining module needs to perform 2 main database-related tasks, insert those *aspects* that have been found by the *aspect* extraction sub-module and update orientations (that were previously set to 0,) according to the orientation finder sub-module.

Simple Insert Aspect, consists of saving three main objects in the database: (1) the list of aspects that has been found, (2) sentences that include any aspect, and (3) the *position* of every *aspect word* in all those sentence where it appears. Orientations on aspects will be set to 0. On the other hand, Insert Aspects (a task that extends Simple Insert Aspect) considers the same three steps, but also includes inserting previously extracted near adjectives for each *aspect* and all the original words that originated each *aspect* word.

Finally, Insert Orientation simply refers to the task of inserting the orientation of each aspect in each sentence, according to the results of the orientation determination process.

## PEM Tasks

PEM database-related tasks are related to three main services:

- Insert information located in the corpus .txt file, including all annotated *aspects* with their respective orientations. Also, a special model regarding the human tagger should be added.
- Insert model data, including saving the parameters of the modes used to extract opinions from a corpus.
- Save opinions, storing in the database all the *aspects* and their corresponding orientations extracted from a corpus by a model.
- Get the data necessary to obtain all the performance measures defined in the previous section.

## RVM Tasks

The Results Visualization Module interacts with the DPM in four ways:

- The task Get Adjectives corresponds to a simple query that obtains those adjectives that appear near an *aspect*, also including frequencies of appearance, as needed to prepare adjective bubble charts.

- Get aspect-based data for summaries, including the list of *aspects* and the orientations of each one in the sentences. In other words, all data needed to build the aspect-based summaries (bar charts) has to be obtained by the DPM and returned to the RVM.
- Another task implies making a query to get opinions (i.e. sentences) regarding an *aspect* or an *entity*, including their respective orientations. The result of this query is sent to the RVM.
- Insert an *Entity* is the only task that involves modifying the database. This task implies taking a CSV file, as given by the user through the RVM, and inserting the data contained in it – the corresponding *entity* and the reviews that appear in the CSV file. Also, each review needs to be split into sentences which are also saved.

# Chapter 5

## Implementation

This chapter explains the software that was developed in order to make clear how all relevant processes, algorithms and specifications were put into practice. This software will follow the models given in Chapter 3 and the design detailed in Chapter 4. It also aims to somehow document all the code that was generated, in order to make it easier to understand and modify for future work.

First, a little explanation about the tools that were selected for development is presented. These tools need to provide all the functions and specifications that make the proposed application possible.

After that, details about the implementation of each module are given. Descriptions about them will include simple class diagrams (adapted from UML notation) and explanations of the key components inside, also indicating how the selected tools are included.

### 5.1 Development Tools

When implementing a software application, it is important to choose an adequate set of tools. These tools must permit the programmer to achieve the proposed goals with relative simplicity, while at the same, let him successfully fulfill the requirements.

Below, the most important tools that were selected are introduced, indicating in each case the reasons that made them the best option to choose.

- **Ubuntu Linux Operating System:** A computer operating system based on the Debian Linux distribution, distributed as free and open-source software, using its own desktop environment. As of 2012, according to on-line surveys, Ubuntu is the most popular Linux distribution on desktop/laptop personal computers, and most Ubuntu coverage focuses on its use in that market. In this case, Ubuntu has been chosen since it presents a stable platform for development, giving users free access to a complete set of fully- documented applications and a very user-friendly interface.

- **Python Programming Language:** A programming language that focuses on allowing people to work more quickly and integrate systems more effectively [101]. Python runs on Windows, Linux/Unix and Mac OS, and is frequently included in most Linux distributions such as Ubuntu. Python is free to use, even for commercial products, because of its OSI-approved open-source license. Some of the reasons that made Python the better choice were:
  - A very clear and readable syntax.
  - Intuitive object orientation.
  - Full modularity, supporting hierarchical packages.
  - Exception-based error handling.
  - Very high level dynamic data types, including built-in dictionaries, lists and similar data structures.
  - Extensive standard libraries and third-party modules, including a very intuitive package manager.
  - Python code is embeddable within applications as a scripting interface.
- **Natural Language Toolkit:** A platform for building Python programs to work with human language data, implementing some of the most common NLP utilities, like text classification, tokenization, stemming, tagging, parsing, and semantic reasoning, also providing easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet and the Penn Treebank. NLTK is a free, open source, community-driven project. Also, NLTK has full on-line documentation and various manual books that introduce users to programming fundamentals alongside topics in computational linguistics. For this work, the books Python NLTK [102] and NLTK Cookbook 2.0 [28] provided the basis knowledge to implement all the NLP tasks.
- **Orange:** A data mining and machine learning software suite, including Python libraries for scripting. Orange is a component-based suite that includes a complete set of modules for data preprocessing, filtering, modeling, model evaluation, and exploration techniques. Orange is distributed free under the GPL. For this work, Orange provides a set of Python modules to implement the data mining processes used to find aspects; in particular, methods for frequent itemset mining.
- **JSON (JavaScript Object Notation):** A lightweight data-interchange format, easy for humans to read and write and easy for machines to parse and generate. JSON is a text format that is completely language independent because it uses conventions that are familiar to most programmers. In simple words, JSON is built on two structures:
  - A collection of name/value pairs. In various languages, this is realized as an object, record, data structure, dictionary, hash table, keyed list or associative array.
  - An ordered list of values. In most languages, like Python, this is realized as an array, vector, list or sequence.

Clearly, all these are universal data structures that practically all modern programming languages support. This makes JSON an ideal data serialization method.

- **PostgreSQL:** an object-relational database management system, implementing the SQL:2008 standard. Postgres is free, open-source software and is available for many



platforms, including Linux, Microsoft Windows and Mac OS. Postgres provides advanced SQL management with complete on-line documentation and direct integration with most Linux-based OS.

- **Django:** an open-source web application framework, written in Python, which follows the model – view – controller architectural pattern. As stated on the official website [103], Django’s primary goal is to ease the creation of complex, database-driven websites. In this case, Django was chosen because of two reasons:
  - An object-oriented database API, which permits the developer to easily deal with simple and complex database operations.
  - Reusability and pluggability of components, which enables rapid development of web applications by implementing the MVC pattern.
- **Google Charts:** A tool that let web application developers create charts from some data and embed it within a web page. Since charts are exposed as JavaScript classes, populating the data is a simple task using the provided tools, client or server side. Many types of charts are supported, currently including line, bar, pie, bubble and radar charts, as well as Venn diagrams, scatter plots, sparklines, maps, google-o-meters and QR codes. All these charts can be easily customized to fit the desired look and feel and are highly interactive, exhibiting events that permit the creation of experiences that are integrated with the page.

## 5.2 Data Persistence Module

The data persistence module was implemented using two different approaches. A first development includes the use of Django’s Database API for the RVM requirements, while the implementation of the PEM database-related operations simply uses Python Postgres packages. This paradigm was chosen since operations that will be performed in the PEM are non-conventional SQL queries that are not well covered by Django, which focuses on operations commonly needed when designing web applications. Nevertheless, implementation of both data models has a lot of similarities.

### 5.2.1 RVM Data Model

The data model for the RVM was built using the Django Database API, using a Postgres server. The model corresponds to a formalization of the structure given in Chapter 4. Tables and columns were declared using specially disposed classes and objects by the Django Database API in a simple Python module. Using a simple command, objects declared are synchronized with the database and objects are mapped to each created table.

Once objects are created and synchronized, the API provides a complete set of methods that handle basic CRUD operations without needing SQL statements through a model. In this manner, database connection is completely handled by a set of Python classes that can be used from any other Python module by simply importing them. Below, a picture of the generated models mapping all the database objects is presented. Also, an example of the E-R model that is automatically generated is presented in figure 5.1. Since the user never

needs to interact directly with the data mode, table and column names do not necessary correspond to the names appearing in the Python class.

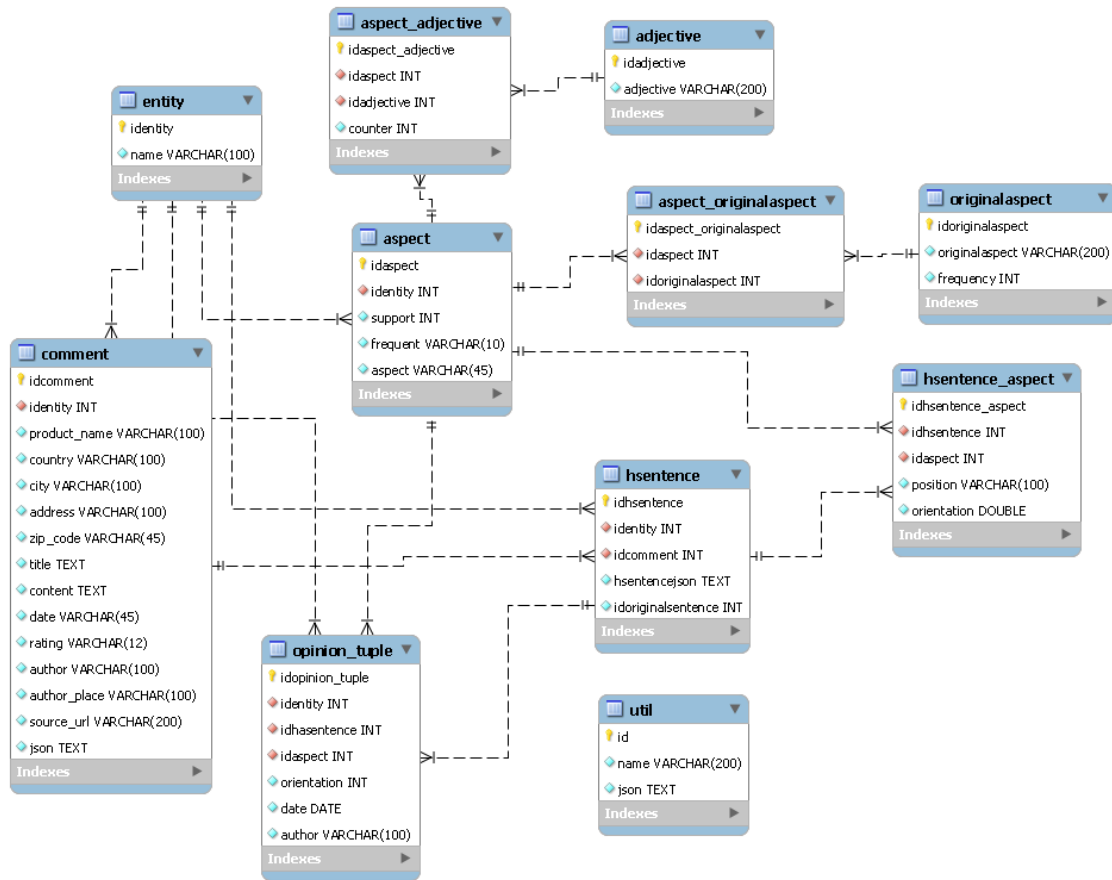


Figure 5.1: E-R Data Model for the RVM.

Source: Own preparation.

Details on each table are given below. The model class can be simply modified to fit any change in the specifications of these tables.

Table entity	
Column	Description
identity (PK)	INTEGER, unique identifier of an entity in the system.
entity	VARCHAR, name of the entity.

Table 5.1: Description of table entity for the RVM Data Model.

Source: Own preparation.

Table comment	
Column	Description
idcomment (PK)	INTEGER, unique identifier of a comment or review in the system.
identity (FK)	INTEGER, foreign key from table entity.
product_name	VARCHAR, name of the product (name of the hotel or restaurant).
country	VARCHAR, country where the hotel or restaurant is located.
city	VARCHAR, city where the hotel or restaurant is located.
address	VARCHAR, address of the hotel or restaurant.
zip_code	VARCHAR, zip code of the corresponding product.
title	TEXT, title of the comment or review.
content	TEXT, title of the comment or review.
date	VARCHAR, date of the moment the comment was posted.
rating	VARCHAR, rating that the user gives to the product in the comment.
author	VARCHAR, name of the author of the review.
author_place	VARCHAR, place (city or county or both) where the author of the review is from (if given.)
source_url	VARCHAR, TripAdvisor URL where the comment was downloaded from.
json	TEXT, JSON object containing the set of sentences that the comment has.

Table 5.2: Description of table comment for the RVM Data Model.  
Source: Own preparation.

Table aspect	
Column	Description
idaspect (PK)	INTEGER, unique identifier of an <i>aspect</i> in the system. As delivered by the aspect expression extraction algorithm, an <i>aspect</i> is a set of stems.
identity (FK)	INTEGER, foreign key, from table entity.
aspect	VARCHAR, name of the aspect (set of stems.)
frequent	VARCHAR, unused.
support	INTEGER, number sentences where the aspect appears.

Table 5.3: Description of table aspect for the RVM Data Model.  
Source: Own preparation.

Table originalaspect	
Column	Description
idoriginalaspect (PK)	INTEGER, unique identifier of an original aspect in the system. An original aspect will correspond to the original word that originated any word of an <i>aspect expression</i> .
originalaspect	VARCHAR, original word of the originalaspect.
frequency	INTEGER, the number of times that the originalaspect originated a word of any aspect in the system.

Table 5.4: Description of table originalaspect for the RVM Data Model.  
Source: Own preparation.

Table aspect_originalaspect	
Column	Description
idaspect_originalaspect (PK)	INTEGER, unique identifier of an idaspect_originalaspect in the system. Each one of them will denote an originalaspect contained in the corresponding <i>aspect expression</i> .
idaspect (FK)	INTEGER, foreign key of table aspect.
idoriginalaspect (FK)	INTEGER, foreign key of table originalaspect.

Table 5.5: Description of table aspect\_originalaspect for the RVM Data Model.  
Source: Own preparation.

Table hsentence	
Column	Description
idhsentence (PK)	INTEGER, unique identifier of an hsentence in the system. An hsentence will correspond to a sentence that has any aspect in it.
idcomment (FK)	INTEGER, foreign key from table comment.
hsentencejson	TEXT, is the JSON object that will
idoriginalsentence	INTEGER, is the original number of the sentence that originated the hsentence, considering the whole set of sentences extracted from all documents processed.

Table 5.6: Description of table hsentence for the RVM Data Model.  
Source: Own preparation.

Table hsentence_aspect	
Column	Description
idhsentence_aspect (PK)	INTEGER, unique identifier of an hsentence_aspect in the system. Each one of these will denote a pair $(ae, s)$ $ae \in AE, s \in S$ .
idhsentence (FK)	INTEGER, foreign key from table hsentence.
idaspect (FK)	INTEGER, foreign key from table aspect.
position	TEXT, correspond to a JSON object that will store the array of the positions of each word of the aspect expression $ae$ in the sentence $s$ .
orientation	REAL, corresponds to the value of the orientation that the Orientation Determination algorithm returns.

Table 5.7: Description of table hsentence\_aspect for the RVM Data Model.  
Source: Own preparation.

Table adjective	
Column	Description
idadjective (PK)	INTEGER, unique identifier of an adjective in the system.
adjective	VARCHAR, the real adjective (word) that is saved in the system.

Table 5.8: Description of table adjective for the RVM Data Model.  
Source: Own preparation.

Table aspect_adjective	
Column	Description
idaspect_adjective (PK)	INTEGER, unique identifier of an aspect_adjective in the system.
idaspect (FK)	INTEGER, foreign key from table aspect, indicating the id of the aspect from which the adjective appears near.
idadjective (FK)	INTEGER, foreign key from table adjective, indicating the adjective appearing near the aspect with id = idaspect.
counter	INTEGER, counts the number of times that the adjective appears near the aspect with id = idaspect.

Table 5.9: Description of table aspect\_adjective for the RVM Data Model.

Source: Own preparation.

Table util	
Column	Description
id(PK)	INTEGER, unique identifier of an util in the system.
name	VARCHAR, name of a util. An util is a serialized variable that permits some RVM operations to be performed.
json	TEXT, serialized variable stored in the database. Once the variable is used, the system will automatically delete it.

Table 5.10: Description of table util for the RVM Data Model.

Source: Own preparation.

## 5.2.2 PEM Data Model

Figure 5.2 presents the E-R model that was used in this case. Details on each table are shown below.

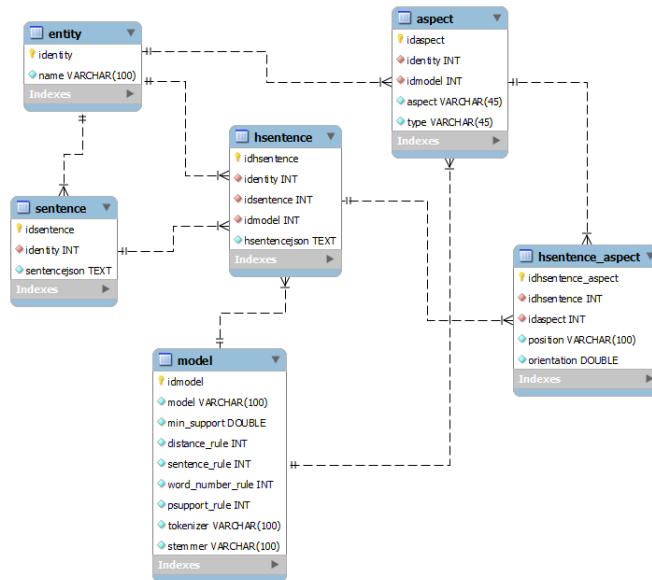


Figure 5.2: E-R Data Model for the PEM.

Source: Own preparation.

Table entity	
Column	Description
identity (PK)	INTEGER, unique identifier of an entity in the system.
entity	VARCHAR, name of the entity. In this case, the entity will denote the name of an annotated corpus that has been added to the database.

Table 5.11: Description of table entity for the PEM Data Model.

Source: Own preparation.

Table model	
Column	Description
idmodel (PK)	INTEGER, unique identifier of an aspect_adjective in the system.
model	VARCHAR, name of a stored model. A model is a set of parameters used to extract opinions from a corpus applying the algorithms.
min_support	DOUBLE, minimum frequency of the extracted itemsets that will be candidates to be <i>aspect expressions</i> .
distance_rule	INTEGER, maximum separation between any adjacent item in a compound <i>aspect expression</i> .
sentence_rule	INTEGER, minimum number of sentences where a compound <i>aspect expression</i> must be compact to avoid being pruned.
word_number_rule	INTEGER, maximum number of items in a compound <i>aspect expression</i> .
psupport_rule	INTEGER, minimum psupport that is needed for the redundancy pruning step.
tokenizer	VARCHAR, name of the tokenizing algorithm used.
stemmer	VARCHAR, name of the stemming algorithm used.

Table 5.12: Description of table model for the PEM Data Model.

Source: Own preparation.

Table sentence	
Column	Description
idsentence (PK)	INTEGER, unique identifier of a sentence in the system.
identity (FK)	INTEGER, foreign key from table entity, indicating the id of the corpus that the sentence belongs to.
sentencejson	TEXT, a JSON object containing the sentence as a set of ordered tokens.

Table 5.13: Description of table sentence for the PEM Data Model.

Source: Own preparation.

Table aspect	
Column	Description
idaspect (PK)	INTEGER, unique identifier of an aspect in the system. As delivered by the Aspect Expression Extraction Algorithm, an aspect is a set of stems.
identity (FK)	INTEGER, foreign key, from table entity.
idmodel(FK)	INTEGER, foreign key from table model.
aspect	VARCHAR, name of the aspect (set of stems).
type	VARCHAR, indicates if the aspect appears in an explicit manner (n), or in an implicit manner (u) in the corpus. The field is not used when saving aspects extracted by the algorithms.

Table 5.14: Description of table aspect for the PEM Data Model.

Source: Own preparation.

Table hsentence	
Column	Description
idhsentence (PK)	INTEGER, unique identifier of an hsentence in the system. An hsentence will correspond to a sentence that has any aspect in it.
idsentence(FK)	INTEGER, foreign key from table sentence.
identity (FK)	INTEGER, foreign key from table entity.
idmodel(FK)	INTEGER, foreign key from table model.
hsentencejson	TEXT, is the JSON object that will

Table 5.15: Description of table hsentence for the PEM Data Model.

Source: Own preparation.

Table hsentence_aspect	
Column	Description
idhsentence_aspect (PK)	INTEGER, unique identifier of an hsentence_aspect in the system. Each one of these will denote a pair $(ae, s)$ $ae \in AE, s \in S$ .
idhsentence (FK)	INTEGER, foreign key from table hsentence.
idaspect (FK)	INTEGER, foreign key from table aspect.
position	TEXT, correspond to a JSON object that will store the array of the positions of each word of the aspect expression $ae$ in the sentence $s$ . This field is not used when saving an annotated <i>aspect</i> in a corpus.
orientation	REAL, corresponds to the value of the orientation that the Orientation Determination algorithm returns or is given by the annotated corpus.

Table 5.16: Description of table hsentence\_aspect for the PEM Data Model.

Source: Own preparation.

All the database-related operations that the PEM requires are implemented using the `pygresql` module, which provides methods to create and manage a database connection, by sending raw SQL queries. In this case, this alternative was preferred since a lot of performance evaluation metrics need complex database queries to be performed. Since Django does not provide any improved tool in this context, a classic approach was then selected. Figure 5.3 shows the structure of the Python class containing all the needed methods.

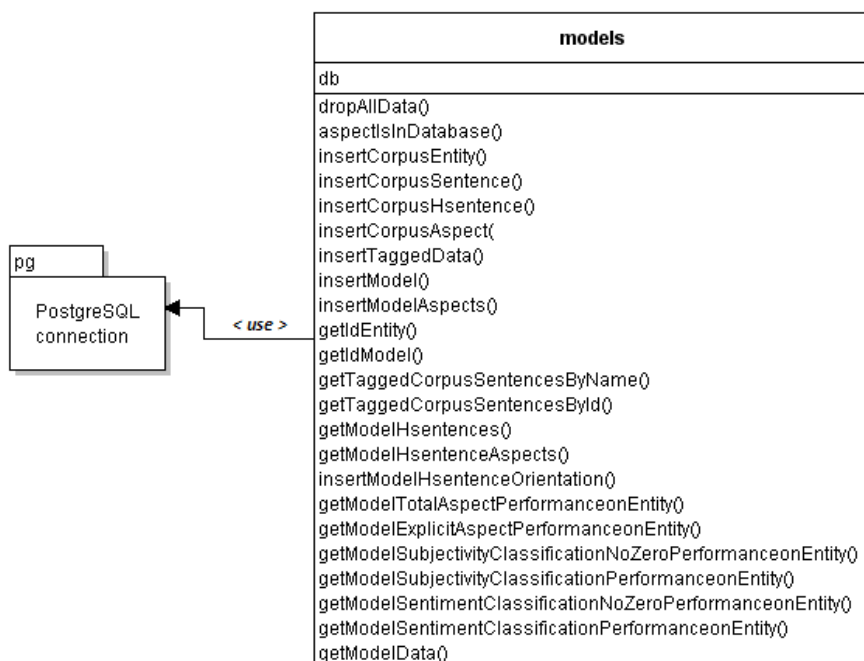


Figure 5.3: Models class diagram.

Source: Own elaboration.

### 5.3 Data Collecting Module

The DCM was implemented using three Python libraries:

- `urllib`: a module that defines functions and classes to help in opening URLs (mostly HTTP), including authentication, redirections and cookies. This module was used in order to get and download the HTML files that contained opinions regarding hotels and restaurants in the Lake District.
- `threading`: module that implements threads in the Python programming language. Threads were used to simultaneously download HTML pages with opinions, reducing the amount of time needed to visit and download all pages.
- `lxml`: a module that provides a full-featured and easy-to-use library for processing XML and HTML in Python. This module is capable of parsing an HTML file as a tree object of nested tags. In this tree, special div classes are recognized and can be easily accessed in order to obtain their content.

These three modules are bundled into two different classes that, so far, must be called one by one (automation is fairly simple but was not necessary here). Figure 5.4 shows a simple class diagram of the whole module.



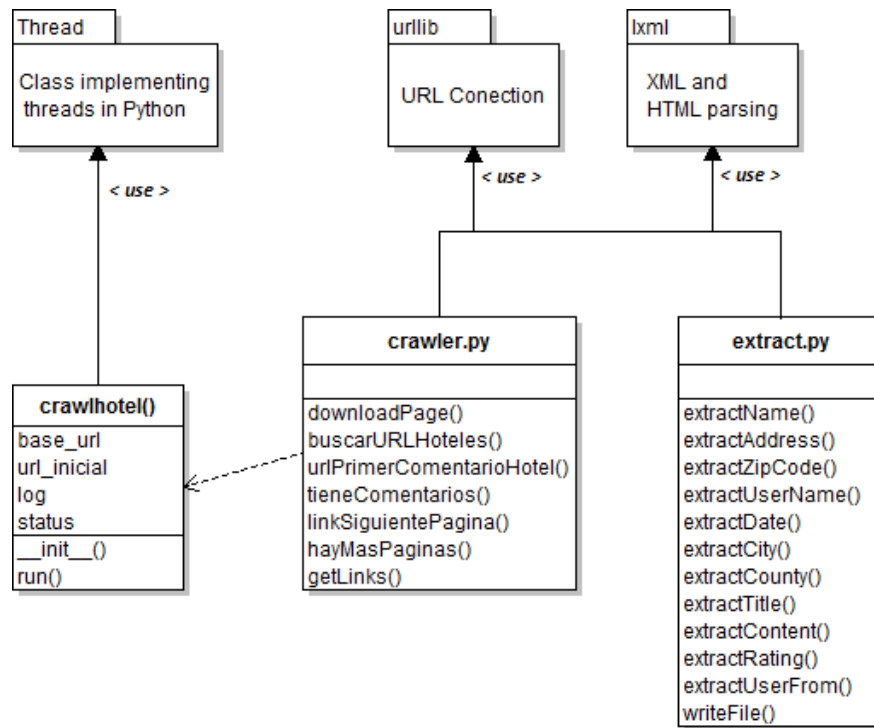


Figure 5.4: DCM class diagram.

Source: Own elaboration.

- crawler.py: The crawler takes as input the TripAdvisor base URL (for the US version) and the URL of the homepage for a *topic*, considering that TripAdvisor URLs are built in the following manner:

$$\boxed{\text{www.tripadvisor.com}} + \boxed{/URL} + \# \text{ TEXT}$$

Once both addresses have been manually identified and passed to the crawler as simple string parameters, it starts downloading all the HTML files that contain opinions using the urllib module. To do this, the program generates one thread for each product (restaurant or hotel) detected in the product's home page and tries to visit and download all the pages with reviews. These pages are detected by parsing specific HTML tags (div classes) that contain links to pages with opinions, using the lxml module. All the files are downloaded in the same path where the program is. Also, the crawler writes a set of log files (in plain text format) containing an index of all the HTML files that were downloaded.

- extract.py: This program reads the log files written by the crawler and starts parsing the downloaded HTML files to obtain all the requested information from each one of them. As when crawling, the parsing process consists of extracting the content from specific HTML tags using the lxml library. These tags were previously manually identified in an exhaustive review of the website. The extracted data is stored in a comma-separated CSV file. In this file, the content of each review needs special attention in order to avoid problems with the comma used to separate fields in the CSV file. For that reason, the content of each review was encapsulated in a special tag (`<comment>` `</comment>`).

### 5.3.1 Data for Corpora

The downloaded CSV files were used to generate the raw (not annotated) data for the evaluation corpora. As the number of data clearly exceeded humanly manipulable size, and it was also considerably greater than the corpora generated by Bing Liu, a random sampling process was implemented in Python, using the *itertools* module, which provides methods to generate pseudo-random numbers and use them to select items from an *iterable* object. The random sampler selected a total of 100 random reviews for each case (hotels and restaurants) and saved the data in a .txt file, according to the structure designed in the last chapter. Original reviews were segmented in sentences using unsupervised machine learning techniques, particularly the *Punkt Sentence Tokenizer* provided by the NLTK libraries.

## 5.4 Opinion Mining Module

This module is in charge of all the NLP and data mining operations that are needed to find aspects and their orientation in each sentence. The following Python packages were used.

- NLTK and Orange
- math: Python module that provides access to the mathematical functions, as defined by the C standard.
- re: Python module for regular expression handling, providing most common matching operations and others similar to those found in Perl Programming Language.
- guess\_language: A Python module that attempts to determine the natural language of a text (in utf-8 format). This module uses heuristics based on trigrams in a sample text, detecting over 60 languages.
- json: A Python module implementing the JSON format. The module provides methods for encoding and decoding Python native objects into JSON format.

The OMM was implemented strictly following the modular programming paradigm. As a result, several classes were generated, each one focusing on a basic set of requirements. These classes include the two sub-modules proposed in the design, but also consider 2 other classes that provide methods and functionalities for them. Below, these two classes are introduced, figure 5.5 shows the classes structure.

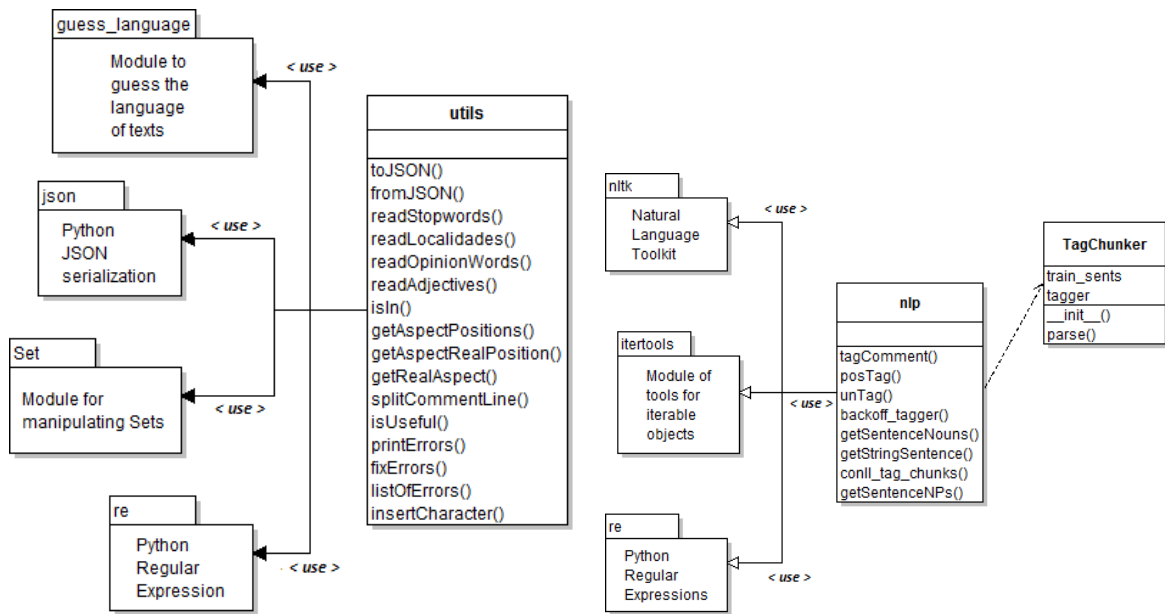


Figure 5.5: Complementary class diagrams for the OMM.

Source: Own elaboration.

- `nlp`: Provides methods related to all NLP tasks needed for the OMM. These tasks include the Punkt Sentence Tokenizer, POS tagging, chunking and others similar. The `nlp` module is almost purely based on the NLTK library, also including some methods for regular expression handling.
- `utils`: A class that provides a set of methods that is used by both, the aspect extraction and the orientation finder sub-modules. These methods include JSON and Set objects handling, text language guessing (based on `guess.language`) and some regular expression handling. In particular, this class offers methods to read the CSV file and process its content by separating each review into sentences using the Punkt Sentence Tokenizer, provided by the `nlp` module. The module also fixes some common punctuation mistakes (e.g. missing periods or missing spaces after periods) in the reviews that, once fixed, greatly increase the performance of the sentence tokenizer.

### 5.4.1 Aspect Extraction Sub-Module

This sub-module is implemented by the Python class called `aspect_extractor`. Therefore, this class offers all the methods needed to implement the *aspect expression* extraction algorithm developed in Chapter 3. The class is structured as figure 5.6 shows. It receives as input a set of reviews and their respective sentences regarding a topic (as requested to the DPM,) and works in the following manner:

- (1) For each sentence of each review:
  - (a) Tokenize a sentence into words.
  - (b) Apply POS tagging to each sentence (as a set of tokens) and extract nouns from the sentence.

- (c) Apply chunking to the POS-tagged sentence and extract the NPs.
  - (d) Generate a list of all extracted nouns and NPs and delete stopwords, tourism related terms, numbers and adjectives.
  - (e) Write a line of a transactional basket file (space and comma-separated words) including all the stems of each noun and NP in the list, using the Porter Stemmer. For each word written in the file, a `sentence_item` object is generated. This object saves the position, the original word, POS tag and the stem of all the written words. Each `sentence_item` of a sentence is saved in a `sentence.items` object which is also added to a list called `sentences_items`. Also, the POS-tagged sentence is added to a list-like object containing all the sentences. The review to which each sentence belongs is saved in an object called `inverse_sentence_index`.
- (2) Once all the sentences have been processed, a frequent itemset mining algorithm is applied to the transactional file using a method provided by the Orange library and a minimum support rule. This rule is then a parameter to the system and needs to be previously set. As a result, an object containing all the frequently-extracted itemsets is generated. This object, called `aspects_list`, is a dictionary-like object containing, for each itemset, the indexes of all the sentences where it appears.
  - (3) Apply Compactness Pruning Rules to `aspects_list`. Three rules need to be set: the maximum number of words in an *aspect expression*, the maximum possible distance between words in a compound *aspect expression* and the number of sentences where a compound *aspect expression* needs to be compact to be considered as an *aspect expression*.
  - (4) Apply Redundancy Pruning Rules to `aspects_list`. One parameter needs to be set: the number of sentences where each sub-aspect must appear to avoid being pruned.
  - (5) Delete non-compact appearances of compound aspects in the object `aspects_list`, since it will be considered that a compound *aspect expression* will only appear in those sentences where it is compact.

As a result, objects are generated containing all the information needed to save the aspects and their respective sentences:

- the object `aspects_list`, containing final list of extracted aspects.
- the object `inverse_sentence_index`
- the object `sentences_items`
- the list indexing all the POS-tagged sentences that were processed.

These objects are passed to the DPM, which executes all the processes needed to save each sentence and corresponding aspect, including their positions.

## 5.4.2 Orientation Finder Sub-Module

This sub-module is implemented by the class `orientation_finder`, which takes as input a set of already extracted aspects and their sentences by requesting a query to the DPM. The

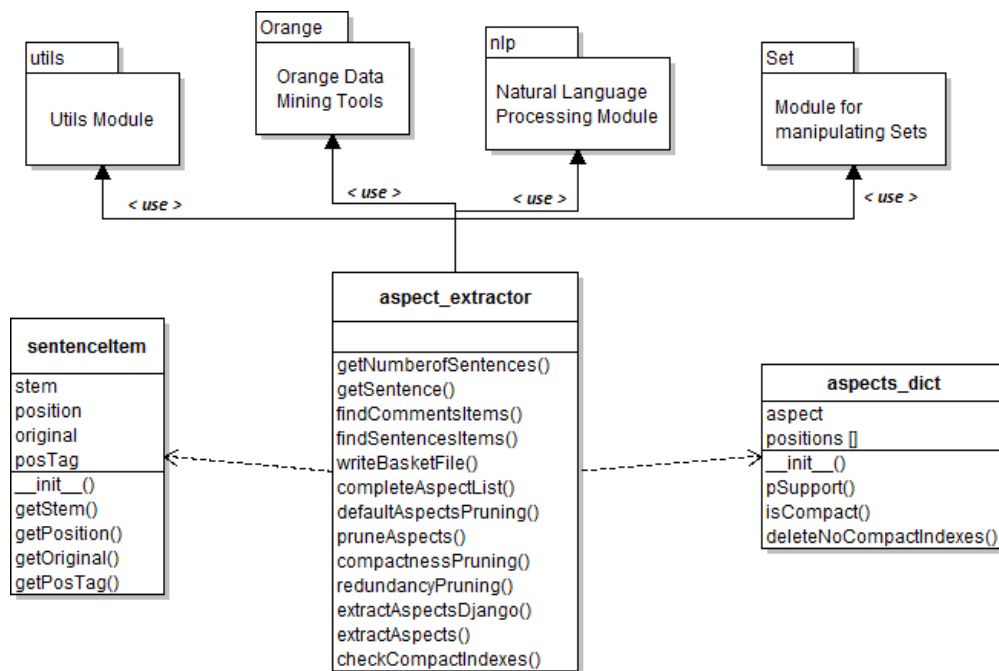


Figure 5.6: Class diagram of the aspect\_extractor package.

Source: Own elaboration.

sub-module is structured according to what is shown in figure 5.7, below.

The Orientation Finder sub-module directly implements the algorithms developed in Chapter 3. The implementation consists of generating a special object that marks each word in a POS-tagged sentence as being affected by a set of rules. Once all the rules are applied, the orientation of each word is obtained using the procedure `word_orientation`, nevertheless some extra specifications were needed. These specifications were added since it was necessary to define a boundary limit for the application of some rules.

- Negation Rule: After any negation words, the following four words will be affected by the rule. This boundary was set purely based on empirical experimentation and intuition.
- Negation Pattern Rule: Since patterns are more sophisticated than simple negation words, the two words appearing before the first word of the negation pattern are affected by the rule. Again, this criterion was intuitive.
- Too Rule: When a *too word* appears, the next two words will be affected by the rule (this is another intuitive criterion).
- Adjective Extraction Rule: the two adjectives nearest to each word of an *aspect expression* are extracted. This rule is also intuitive.

When the final orientation of one aspect in a sentence has been determined, the sub-module sends a request to the DPM to update the orientation previously set to 0.

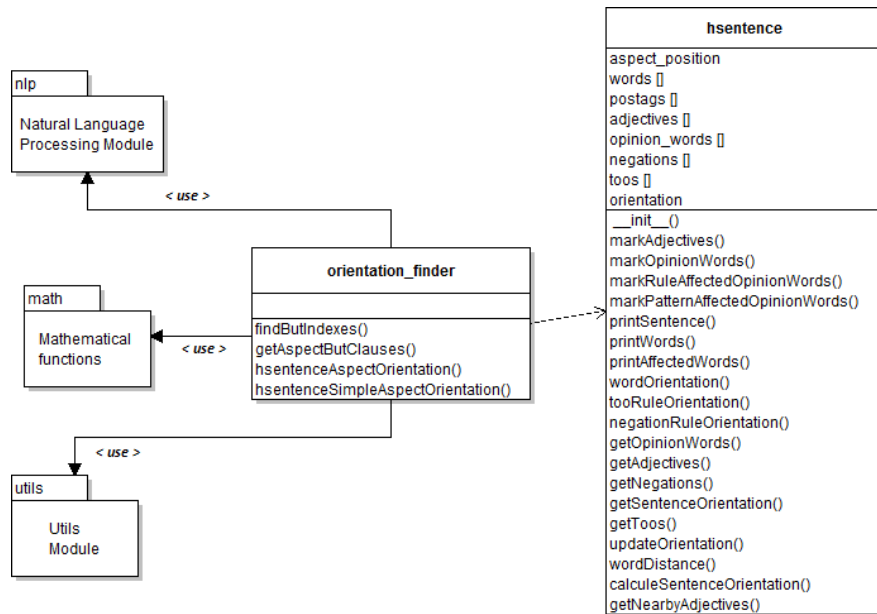


Figure 5.7: Class diagram of the orientation\_finder class.

Source: Own preparation.

## 5.5 Performance Evaluation Module

The PEM does not require a special client-oriented visualization layer since it will only be managed by an experienced special user. For that reason, the PEM was conceived as a Python API-like application, implemented as a simple class called performance\_evaluator. In this simple application, all the processes are requested by running a specific method after importing the performance\_evaluator module in the Python Interactive Console. Figure 5.8 shows how the class is structured.

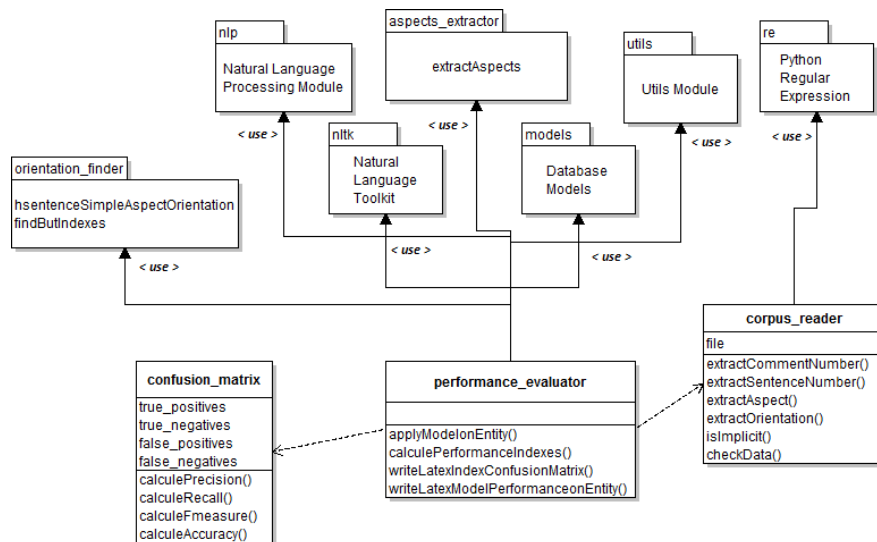


Figure 5.8: PEM class diagram.

Source: Own elaboration

## 5.6 Results Visualization Module

The RVM was implemented using a simple web application, developed with the Django Framework. For this work, the RVM is presented as an initial proof of concept in order to demonstrate the potential of the ideas proposed. Since a proof of concept is usually small and may or may not be complete, the application will not fulfill all the requirements given in the last chapter and will not consider design concerns, focusing on showing the most important functionalities.

Since Django is purely implemented in Python and follows the MVC pattern, the integration of the RVM with other modules is direct. Details on implementation will not be given here since most of the programmed functions and classes obey the structure that Django imposes. However, a little explanation of this structure is relevant since it helps to understand how the module integration was achieved. Django proposes some core components:

- Models and Database API, as used in the DPM.
- Templates: A set of classes that are in charge of generating HTML files according to specific inputs. Django implements template language, designed to feel comfortable to those used to working with HTML. Nevertheless, the Django template system is not just Python embedded into HTML; it is meant to express presentation, not program logic. A template is simply a text file, generated by any text-based format (HTML, XML, CSV, etc.,) containing variables that get replaced with values when the template is evaluated, and tags, which control the logic of the template and function similarly to some programming constructs. In the application, templates were built in the simplest possible manner.
- Views: A type of Web page in a Django application that generally serves a specific function and has a specific template. Views are functions of the main class and are in charge of implementing specific tasks or processes that need to be carried out by the application.
- URLs: A unique identifier of a piece of Python code. URLs are defined in a Python module called URLconf, which associates a given URL with given Python code, generally a view. URLs saved in the URLconf module are written in the manner of regular expressions. When a user requests a Django-powered page, the system looks for the requested URL inside the URLconf module by searching to find if it fits any specific regular expression. When the system finds a match, Django calls the corresponding function.

Given this MVC paradigm, integrating functionalities that the OMM and DCM provide to a Django Web Application is fairly simple. Python's modular programming paradigm permits it to simply inherit specific functionalities of a particular class by calling the *import* statement. Then, by simply importing the needed functions from the corresponding modules, all the processes that are run by these modules can be called from any class inside the RVM (the Django application.) Figure 5.9 shows the structure of the developed web application.

Aspect-based summary bar charts and adjective bubble charts are provided by the Google Charts API. Charts can be adapted to multiple visualization styles but here only basic features were used. On the other hand, users can explore their content by triggering events that are managed by the API, which make charts interactive. Since any JavaScript function can be called when an event is triggered, charts can interact with users in almost any imaginable way. Below, some proposals about possible user-chart interaction events are given. Some of these proposals were implemented, but because of a lack of time, some others were not.

- When a user clicks an *aspect* bar in the aspect-based summary chart, a pop-up window offers to visit a page where the summary for that aspect (the adjective bubble chart) is presented.
- Alongside the bar chart, a table with the *positive score*, *negative score* and *relative importance* of each *aspect* is shown. By clicking the title of each column, the table and the bar chart will sort according to the clicked value.
- When a user clicks an adjective in the bubble chart sentences that contain that adjective and the corresponding aspect are shown.
- When a user clicks a sentence details about the opinion holder or original review are shown.



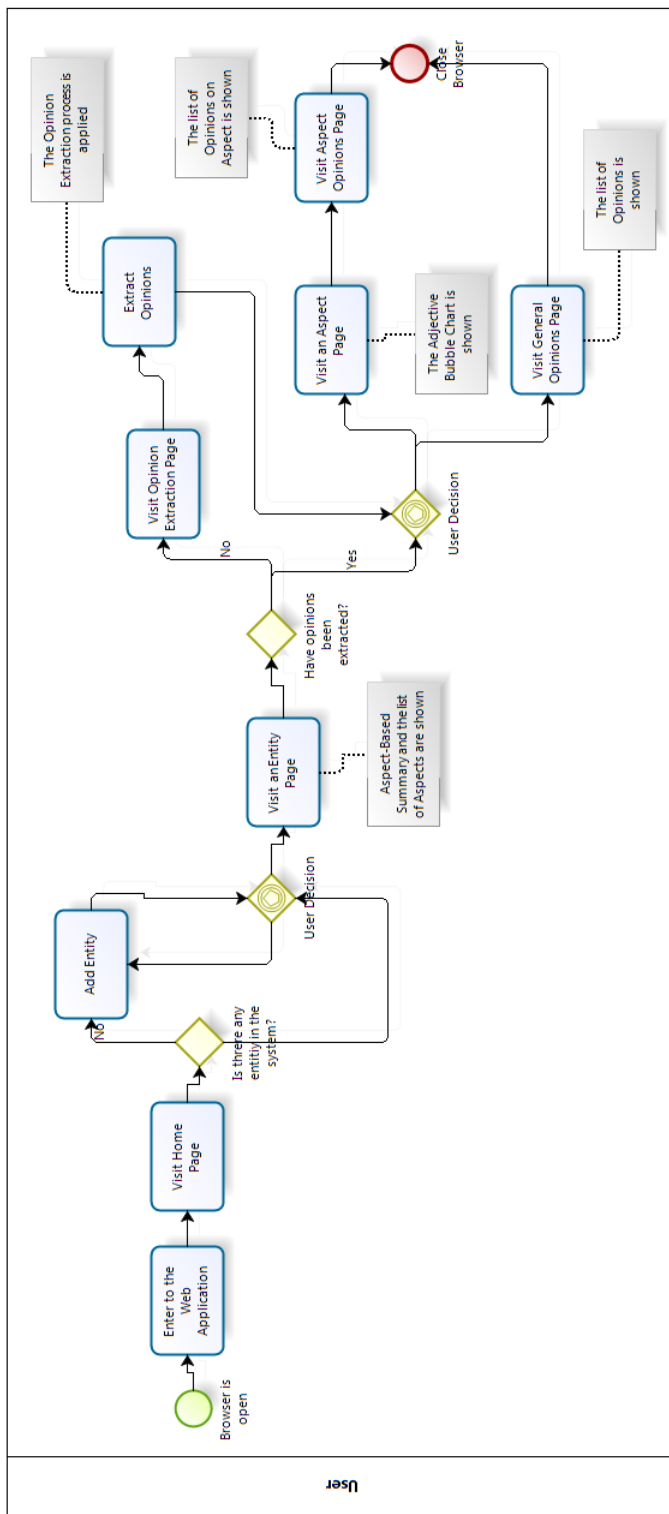


Figure 5.9: Website structure design.

Source: Own elaboration.

# Chapter 6

## Results

In this chapter, all the results obtained from the implementation of the proposed application are shown in a simple and ordered manner. First, a description of the annotated corpus that was elaborated is presented, also commenting on some details in relation to the elaboration process.

After that, results of opinion mining algorithm performance are presented. Performance measures are shown and obtained values are discussed separately for the hotels and restaurants corpora. Also, a comparison between results obtained here and by Bing Liu is included. Some reasons that could have made the results of this study better or worse than Liu's are explored too.

Finally, features of the implemented application are shown, including images of the most important functionalities and some commentaries about usability and the problems experienced during implementation.

### 6.1 Annotated Corpora

Following the tagging rules, the two files generated in the implementation stage were painstakingly annotated. The annotation process was very time-consuming and requested a lot of attention from the tagger. In the beginning, rules were difficult to apply because there were a lot of cases that were not clearly covered. However, experience helped in developing new rules that covered more cases. Near the end, the process became easier.

*Aspect expressions* that included typos or misspellings were tagged without fixing these mistakes (i.e. exactly as they appeared in the text), since any opinion mining algorithms should be able to extract *aspect expressions* regardless they are misspelled words. Spell checking is considered to be a completely different task from aspect extraction, and although it is important accounting on what is proposed in Chapter 2, it will not be evaluated here.

On the other hand, sentences that seemed ambiguous or really difficult to tag were di-

scussed with a second human annotator, an expert in linguistics. Once an agreement was achieved, the sentence was tagged according to that agreement. This marks an important difference between this study and other tagging procedures commonly carried out in literature, where different annotators tag the same corpus separately and only once the annotation procedure has finished are different results of the same corpus compared. A different approach was used here due to time constraints, since it seemed more efficient and was worth trying as a contribution to research in this field.

As Liu has also stated, considering the difficulties that the tagging process presented, errors and inconsistencies are inevitable, even with the help of the rules. On the other hand, since taggers are humans, subjectivity, although avoided, is also inevitable. Details about the obtained results are presented separately for hotels and restaurants in the following sections.

### 6.1.1 Annotated Hotels Corpus

The tagging process for the hotels corpus was the most complicated because of the complexity of some sentences. This task was also more time-consuming, since it considered a higher number of sentences. Table 6.1 shows some general details about the corpus.

Hotels Corpus	
Number of Reviews	100
Total Number of Sentences	789
Sentences / Reviews	7,89
Number of Opinion Sentences	609
Opinion Sentences/Reviews	6,09
Opinion Sentences/Sentences	77,19%

Table 6.1: Hotels Corpus Details.

Source: Own elaboration.

As table 6.1 shows, the corpus considered a total of 789 sentences in 100 reviews, which means that each review is composed of almost 8 sentences on average. A detailed revision of the corpus showed that the standard deviation of the number of sentences for each review was 5.4. The longest comment had 31 sentences, while the shortest had only 1. As experience during the tagging process showed, this might be showing that while some users tend to *tell a story* about the hotel they visited, some others just limit themselves to giving a precise negative or positive feeling toward a specific situation or *aspect*. Figure 6.1 shows the best-fitting statistical distributions for the number of sentences in this corpus, where the Erlang distribution (continuous) seems to be the best-fitting model. Since the Erlang distribution corresponds to the sum of  $k$  independent identically distributed random variables, each having an exponential distribution, it follows that the number of sentences that a user will write would follow an exponential distribution. On the other hand, the Negative Binomial distribution seems to be the best fitting for the discrete case. However, interpretation here is not direct.

On the other hand, almost 80% of the sentences contained opinions. This shows that, opinionated sentences represent an important fraction of the total sentences, which somewhat validates the use of TripAdvisor as a source of opinions for hotels. Nevertheless, non-

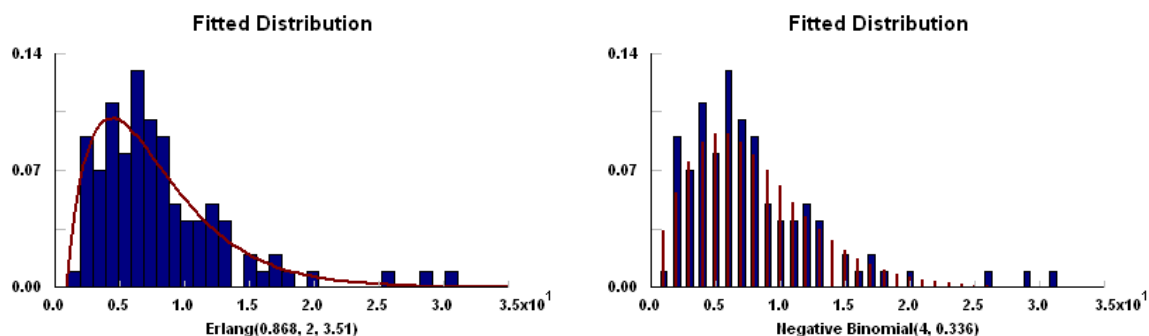


Figure 6.1: Best fitting statistical distributions (continuous and discrete) for the number of sentences in hotels reviews.

Source: Own elaboration using Statfit.

opinionated sentences are also a considerable number, consequently introducing noise into the opinion-extraction process.

Aspect Expressions in Hotels Corpus		
Type	Number	Percentage
Explicit	229	73,87%
Explicit and Implicit	30	9,68%
Implicit	51	16,45%
Total	310	100%

Table 6.2: *Aspect expressions* from the hotels corpus.

Source: Own elaboration.

In relation to the *aspect expressions* that were extracted, table 6.2 shows details about three possible categories. Explicit *aspect expressions* are the most common, representing almost 74%. Also, note that some aspects appear both in an explicit and implicit manner. Since they are explicit expressions in at least one case, they were considered as explicit *aspect expressions*. On the other hand, almost 17% of the extracted *aspect expressions* were purely implicit. A simple review shows that most of these aspects were indicated by adjectives that appear near a special noun. Special attention should be paid to adjectives denoting age (old, new), cleanliness (dirty, clean), size (big, small, spacious), comfort (comfortable), speed (fast, slow) and price (expensive, cheap.)

From these results, it follows that apparently, users tend to use nouns or noun phrases to write reviews, which might be indicating that the developed algorithm should be able to extract a high percentage of the total *aspect expressions*. Also, since adjectives seem to be another common choice to denote *aspects*, they will probably represent the most important downside of the algorithm.

### 6.1.2 Annotated Restaurants Corpus

The tagging process for the restaurants corpus was simpler. As table 6.3 shows, the number of sentences on each review was lower than in the case of hotels, counting a total of 470 sentences and approximately 5 sentences/review. The standard deviation was 2.97 sentences, the longest review contained 17 while the shortest only 1.

Restaurants Corpus	
Number of Reviews	100
Number of Sentences	470
Sentences / Reviews	4,7
Number of Opinion Sentences	368
Opinion Sentences/Reviews	3,68
Opinion Sentences/Sentences	0,78297872

Table 6.3: Restaurants Corpus Details.  
Source: Own elaboration.

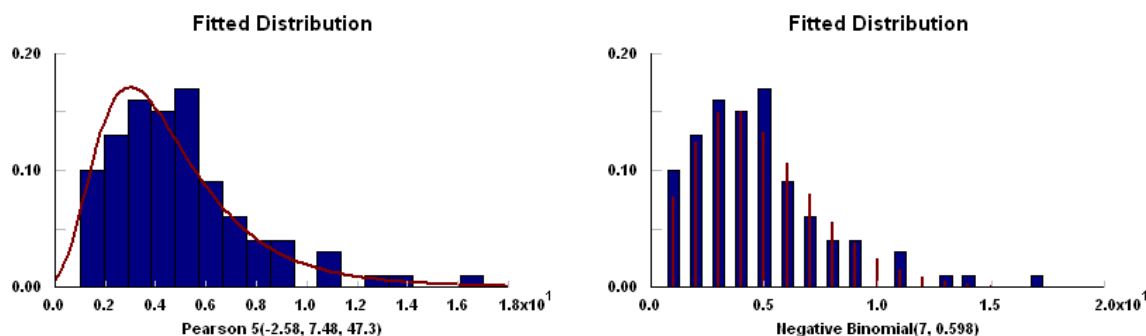


Figure 6.2: Best fitting statistical distributions (continuous and discrete) for the number of sentences in restaurants reviews.

Source: Own elaboration using Statfit.

For this corpus, rules seemed to cover more cases during the whole process. A detailed revision of the corpus shows that for restaurants, users tend to be very precise in telling their *experience*, mentioning dishes they ate and details about the service and the place they visited. However, the proportion of opinionated sentences (78%) is very similar to the hotels corpus, also indicating that non-opinionated sentences might interfere with the opinion extraction process in this case.

Figure 6.2 above shows the best-fitting statistical distributions for the number of sentences in this corpus. In this case, the best model was clearly the Negative Binomial distribution. On the other hand, table 6.4 gives details about the extracted *aspect expressions*.

Aspect Expressions in Restaurants Corpus		
Type	Number	Percentage
Explicit	161	67,93%
Explicit and Implicit	26	10,97%
Implicit	50	21,10%
Total	237	100%

Table 6.4: *Aspect expressions* from the restaurants corpus.  
Source: Own elaboration.

In this case, a little higher percentage of *aspect expressions* appeared in an implicit manner. A revision of the corpus showed that, as in the hotels corpus case, these expressions appeared as adjectives, being the nouns taste (tasty), size (big, small), price (expensive), freshness (fresh) and temperature (hot, cold) the explicit *aspect expressions* that were denoted the most. On the other hand, explicit *aspect expressions* mostly correspond to all kinds

of foods and dishes served in restaurants. Since they are all nouns or noun phrases, the algorithm should be able to extract them with high recall.

## 6.2 Algorithm Performance Evaluation

First, in this section the best general performance of the opinion mining algorithms is presented for each corpus. The best performance was obtained, for each case, by doing a sensitivity analysis regarding the most sensitive parameter - the minimum support rule to extract *aspect expressions*. The six performance measures presented in Chapter 4 were obtained for six different values of the minimum support rule (0.5%, 0.6%, 0.7%, 0.8%, 0.9% and 1%) and the best model was chosen using f-measure values for all measures.

After, average results obtained here are compared with the performance obtained by Bing Liu's algorithms, selecting in each case a comparable metric from his work. Finally, the average performance obtained on both corpora is discussed in general terms.

### 6.2.1 Performance on the Hotels Corpus

Figure 6.3 shows the sensitivity analysis of the six evaluation metrics on the hotels corpus. The best performance was achieved by the model considering the parameters below. Details on the evaluation measures of this model are presented in table 6.5.

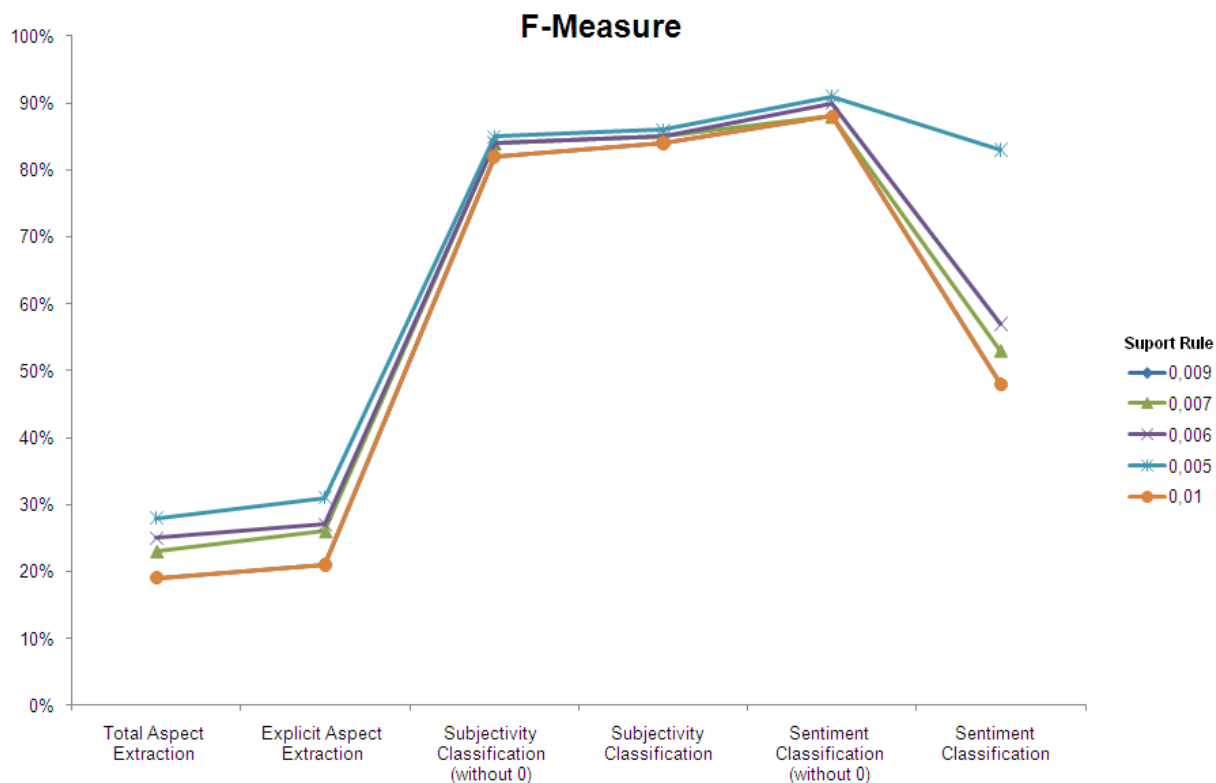


Figure 6.3: Sensitivity analysis on the hotels corpus.

Source: Own elaboration.

- Word Number Rule: 3
- Tokenizer: Treebank Word Tokenizer
- Distance Rule: 3
- Psupport rule: 3
- Stemmer: Porter Stemmer
- Support Rule: 0.005
- Sentence rule: 2

<b>Hotel Model Using Support Rule 0.005</b>				
Index Name	Accuracy	Precision	Recall	F-Measure
Total Aspect Extraction	0.17	0.34	0.25	0.28
Explicit Aspect Extraction	0.18	<b>0.33</b>	<b>0.29</b>	0.31
Subjectivity Classification (without 0)	0.74	0.79	0.9	0.85
Subjectivity Classification	0.76	<b>0.79</b>	<b>0.93</b>	0.86
Sentiment Classification (without 0)	0.85	<b>0.89</b>	<b>0.93</b>	0.91
Sentiment Classification	0.74	0.83	0.84	0.83

Table 6.5: Performance summary of the best model on the hotels corpus.  
Source: Own elaboration.

## 6.2.2 Performance on the Restaurants Corpus

Figure 6.4 shows the sensitivity analysis of the six evaluation metrics on the restaurants corpus. The best performance was achieved by the model considering the parameters indicated below. Details on the evaluation measures of this model are presented in table 6.6.

- Word Number Rule: 3
- Tokenizer: Treebank Word Tokenizer
- Distance Rule: 3
- Psupport rule: 3
- Stemmer: Porter Stemmer
- Support Rule: 0.005
- Sentence rule: 2

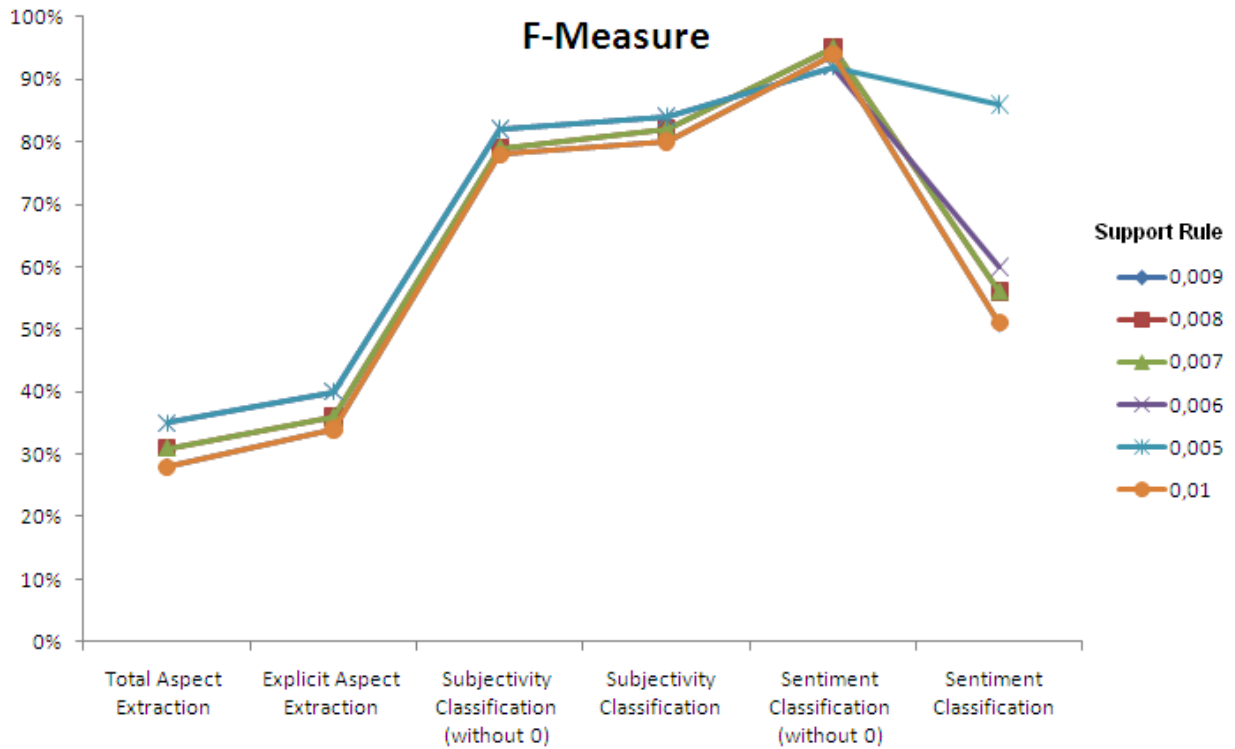


Figure 6.4: Sensitivity analysis on the restaurants corpus.

Source: Own elaboration.

Model Using Support Rule 0.005				
Index Name	Accuracy	Precision	Recall	F-Measure
Total Aspect Extraction	0.21	0.42	0.29	0.35
Explicit Aspect Extraction	0.25	<b>0.42</b>	<b>0.37</b>	0.4
Subjectivity Classification (without 0)	0.71	0.81	0.83	0.82
Subjectivity Classification	0.74	<b>0.81</b>	<b>0.88</b>	0.84
Sentiment Classification (without 0)	0.86	<b>0.91</b>	<b>0.93</b>	0.92
Sentiment Classification	0.77	0.88	0.84	0.86

Table 6.6: Performance summary of the best model on the restaurants corpus.

Source: Own elaboration.

### 6.2.3 Average Performance

Average performance results were computed by simply averaging the results obtained by the best model on both corpora. Table 6.7 compares these results with the performance obtained by Bing Liu.



Index Name	Precision		Recall		F-Measure	
	Here	B. Liu	Here	B. Liu	Here	B. Liu
Explicit Aspect Extraction / P-support pruning [49]	38%	79%	33%	67%	36%	73% *
Subjectivity Classification / Opinion Sentence Detection [49]	80%	64%	91%	69%	85%	67% *
Sentiment Classification (without 0) / Sentiment Classification [64]	90%	91%	93%	90%	92%	90%

Table 6.7: Average performance obtained in both corpora, compared with Liu’s results.

\* Values not directly given by Liu, obtained using Precision and Recall.

Source: Own elaboration using [49] and [64].

Bearing that in mind, these results show that performance on the *aspect extraction* task is fairly poor in the tourism domain. The algorithm is only capable of extracting almost 30% of the total explicit expressions for hotels and almost 40% for restaurants. Moreover, a high percentage of the extracted expressions do not correspond to real *aspect expressions* for both cases. These results tell two important things:

- When users write comments about hotels or restaurants, they mention many objects (i.e. nouns or NPs) that do not correspond to attributes or components of the hotel or restaurant. In other words, users probably *tell stories* about their experiences when writing reviews about tourism products. This may explain the low precision obtained for the explicit *aspect extraction* task in both cases. For instance, in the case of hotels, users commonly refer to objects like time, day, city, which, although relevant for stories, tell nothing about the hotel.
- Users tend to use many different expressions to refer to the same component or attribute of a hotel or restaurant. In this manner, since many different *aspect expressions* are used, most of them will appear in low frequencies in the text. Consequently, items that do not occur with relative high frequencies need to be considered in order to be extracted. In [49], Liu proposed a method to extract these infrequent *aspect expressions* by exploiting their relationships with frequent *opinion words*. In this study, this method was not considered since in Liu’s case, the extracted infrequent *aspect expressions* only represented an improvement of 15% for recall, at the cost of decreasing precision in almost 7%. However, given the poor results that have been obtained, it seems interesting to evaluate how this step improves or worsens performance in this case.

On the other hand, an important improvement in relation to the task of extracting subjective sentences can be noticed. In Liu’s case, the average recall of opinion sentence extraction is nearly 70%, while the average precision of the same task is 64%. Here, although precision increased by 10%, the most important improvement is in recall, in this case 25% higher.

As Liu states in [49] and as seen in previous sections, the reason that explains precision still being a little lower than recall is the fact that users like to tell stories about experiences they lived in relation to products (hotels or restaurants.) In these stories, there are some sentences that do not contain opinions. Since the application labels some of these sentences as opinion sentences because they contain both product *aspect expressions* and some *opinion*

words, precision decreases. Nevertheless, although these sentences may not show strong user opinions towards the product features, they may still be beneficial and useful [49].

Furthermore, the important improvement in recall is probably related to the fact that in [49] Liu and Hu use a very simple algorithm and a small lexicon to perform sentiment classification. Since a more complete approach and a larger opinion lexicon were used here, recall should naturally increase. Nevertheless, the amount of the average recall (90%) obtained here also tells that, although there are a lot of sentences that contain no opinions on the reviews, most of the sentences that really are opinions contain some *aspect expression* and some *opinion word*. This somewhat proves that the definition of a subjective sentence given in Chapter 4 seems fairly acceptable for the tourism domain.

Finally, sentiment classification shows a very little improvement in this work, being a 2% higher than in Liu's case. In this case, most of the possible conclusions are difficult to prove. The increase may be due to the new aggregation functions that were introduced here. Nevertheless, remember that on this case, sentiment classification was only evaluated for those *aspect expressions* that were extracted by the application, while Bing Liu evaluates his algorithm for all the *aspects expressions* that were annotated in his corpora. Since *aspect expressions* that were extracted here are somewhat the simplest ones, determining the sentiment orientation on these aspects may be easier. Consequently, precision and recall could decrease when all *aspect expressions* are considered.

However, the developed algorithm seems to be fairly effective for determining orientations of the *aspect expressions* that were extracted. Evaluating how the algorithm performs considering all the *aspect expressions* appearing in the tourism corpora presented here seems a very interesting direction for future work.

## 6.3 Application Features

First, the application shows the user a home page, where existing entities in the system appear. Also, the option to add a new entity is displayed below. A screen shot of the home page is attached in figure 6.5.

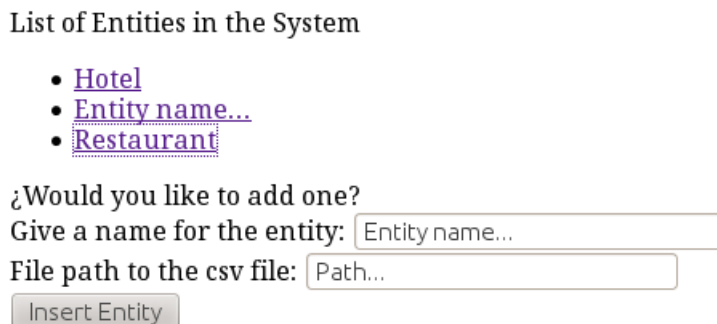


Figure 6.5: Home page of the developed web opinion mining application.

Source: Own elaboration.

The user that wants to add a new entity to the system just needs to type the desired

name for the entity and the path to the CVS file containing the reviews. After clicking on the button *Insert Entity* and waiting for the process to finish, the inserted entity will appear on the list above.

When the user clicks a name in the list of entities displayed on the home page, he is redirected to a special page that shows the aspect-based summary for that entity, in case the opinion extraction process has previously been executed. If when the user clicks an entity name on the home page, opinions have not been extracted yet for that entity, a special menu offering to extract opinions is presented to the user. Figure 6.6 shows details about this process.

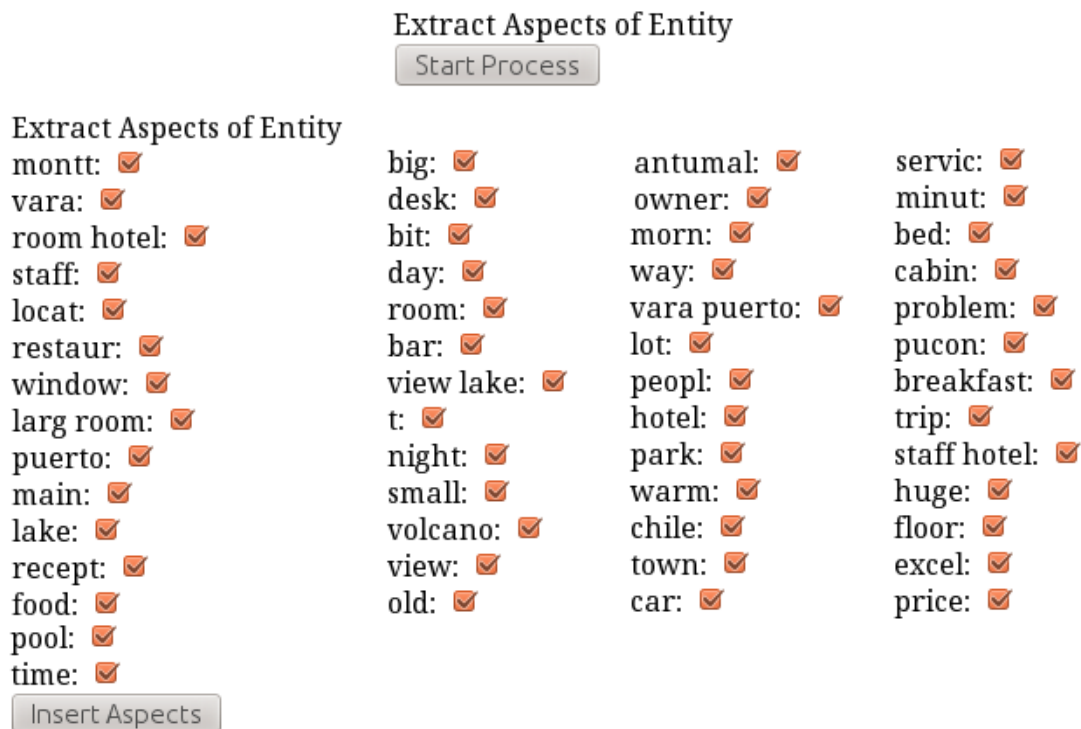
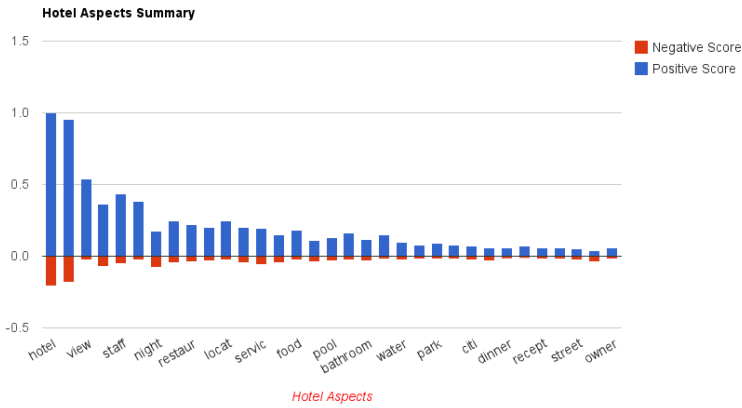


Figure 6.6: Screen shots of the process of extracting and inserting opinions for an entity in the developed application.

Source: Own elaboration.

As the figure indicates, once the applications extract all the *aspect expressions*, since the proposed algorithm presents low precision, a special menu permits the user to select those *aspect expressions* that he wants to save. In case the user is not interested in doing the manual selection, he only needs to click the button *Insert Aspects*, since all the expression will be marked to be saved by default. Once all the aspect expressions are inserted with their corresponding orientations, the user is redirected to the page showing the aspect-based summary. Pictures in figure 6.7 (in next page) show how these summaries are presented for hotels and restaurants.

Aspects

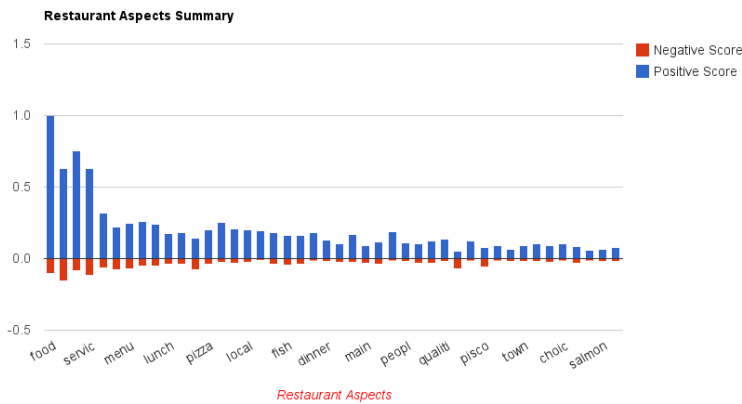


Aspect	Positive Score	Negative Score	Relative Importance
hotel	100%	-21%	100%
room	96%	-18%	97%
view	54%	-3%	64%
breakfast	37%	-7%	37%
staff	43%	-5%	48%
lake	38%	-2%	45%
night	17%	-8%	13%
place	25%	-5%	26%
restaur	22%	-4%	23%
town	20%	-3%	21%
locat	25%	-2%	28%
area	20%	-4%	20%
servic	20%	-6%	18%
stay	15%	-5%	13%
food	18%	-3%	19%
time	11%	-4%	10%
pool	13%	-3%	13%
bed	16%	-3%	18%
bathroom	12%	-3%	11%
volcano	15%	-2%	17%
water	9%	-3%	9%

See details of each aspect:

- [lake](#)
- [time](#)
- [owner](#)
- [breakfast](#)
- [staff](#)
- [bathroom](#)
- [locat](#)
- [restaur](#)
- [citi](#)
- [window](#)

Aspects



Aspect	Positive Score	Negative Score	Relative Importance
staff	25%	-3%	28%
seafood	21%	-4%	22%
local	20%	-3%	23%
coffe	20%	-1%	24%
owner	18%	-4%	19%
fish	16%	-4%	16%
salad	16%	-4%	16%
steak	18%	-1%	22%
dinner	13%	-2%	15%
small	11%	-3%	12%
beer	17%	-3%	19%
main	9%	-4%	9%
waiter	12%	-4%	11%
atmosph	19%	-1%	22%
peopl	11%	-2%	13%
sauc	10%	-3%	11%
meat	12%	-3%	13%
qualiti	13%	-2%	16%
order	5%	-7%	5%
tast	12%	-1%	15%
niern	8%	-6%	8%

See details of each aspect:

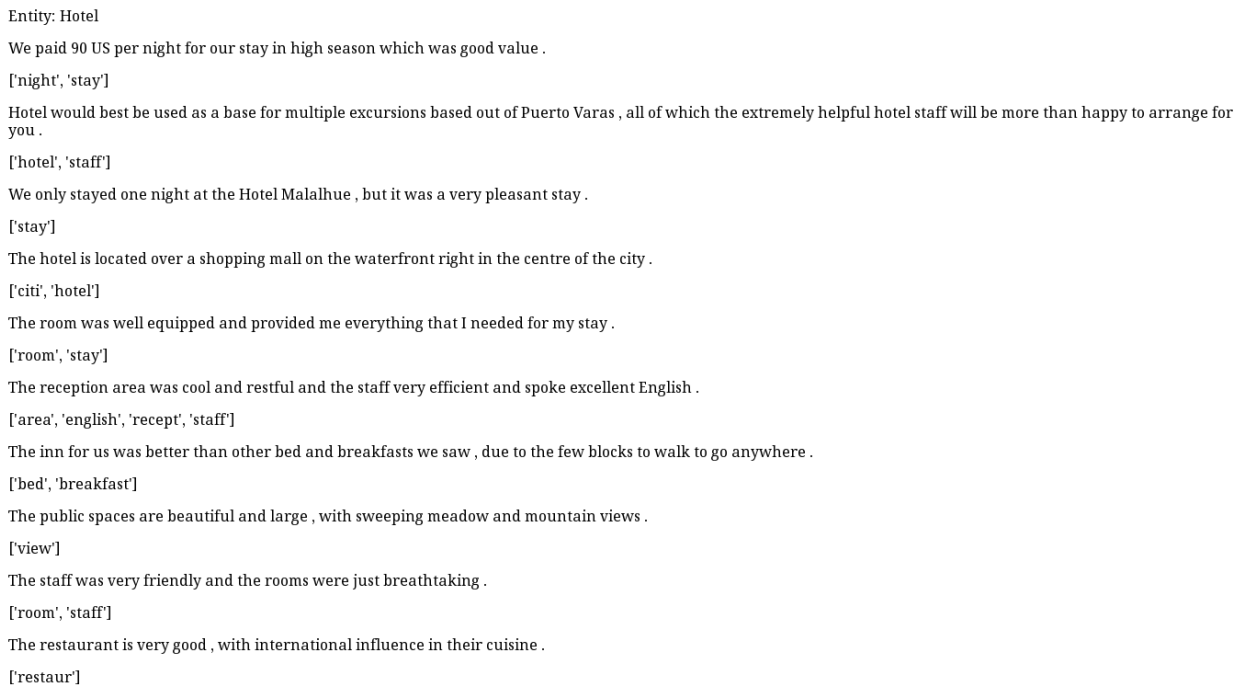
- [atmosph](#)
- [fish](#)
- [tabl](#)
- [qualiti](#)
- [wine](#)
- [restaur](#)
- [seafood](#)
- [owner](#)
- [dish](#)
- [special](#)
- [staff](#)

Figure 6.7: Screen shot of the page showing aspect-based summaries for hotels (above) and restaurants (below).

Source: Own elaboration.

As seen, aspect-based summaries include proposed bar charts and a table that shows the actual values of the *Positive Score*, *Negative Score* and *Relative Importance* for each *aspect expression*. By clicking the name of each column, the table and the bar chart are sorted according to the clicked column (each click alternates between an ascending or descending sort.) Also, below the chart and the table, a list with all the *aspect expressions* is shown. By clicking one of these aspects, the user is redirected to a page showing specific information on that *aspect expression*. Screen shots in figure 6.9 (next page) offer an example of what a user can find in these special pages for *aspect expressions*.

Special pages for *aspect expressions* include adjective bubble charts, as introduced in Chapter 4, and a link to see a list of all opinions on those aspect expressions. These opinions are presented as a simple list of sentences, separated into positive and negative ones. The aspect-based summary page also offers a link to see opinions on the corresponding entity. These opinions are presented as a list of positive and negative sentences. For each sentence, a list of the extracted *aspect expressions* is also shown below. Figure 6.8 below shows an example of positive opinions for the entity hotel.



Entity: Hotel

We paid 90 US per night for our stay in high season which was good value .  
[ 'night', 'stay' ]

Hotel would best be used as a base for multiple excursions based out of Puerto Varas , all of which the extremely helpful hotel staff will be more than happy to arrange for you .  
[ 'hotel', 'staff' ]

We only stayed one night at the Hotel Malahue , but it was a very pleasant stay .  
[ 'stay' ]

The hotel is located over a shopping mall on the waterfront right in the centre of the city .  
[ 'citi', 'hotel' ]

The room was well equipped and provided me everything that I needed for my stay .  
[ 'room', 'stay' ]

The reception area was cool and restful and the staff very efficient and spoke excellent English .  
[ 'area', 'english', 'recept', 'staff' ]

The inn for us was better than other bed and breakfasts we saw , due to the few blocks to walk to go anywhere .  
[ 'bed', 'breakfast' ]

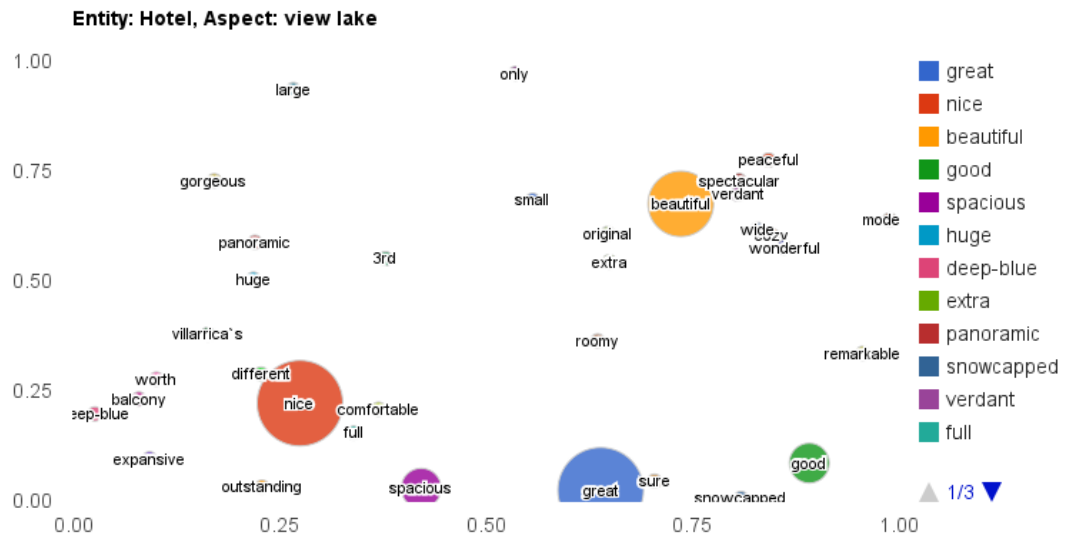
The public spaces are beautiful and large , with sweeping meadow and mountain views .  
[ 'view' ]

The staff was very friendly and the rooms were just breathtaking .  
[ 'room', 'staff' ]

The restaurant is very good , with international influence in their cuisine .  
[ 'restaur' ]

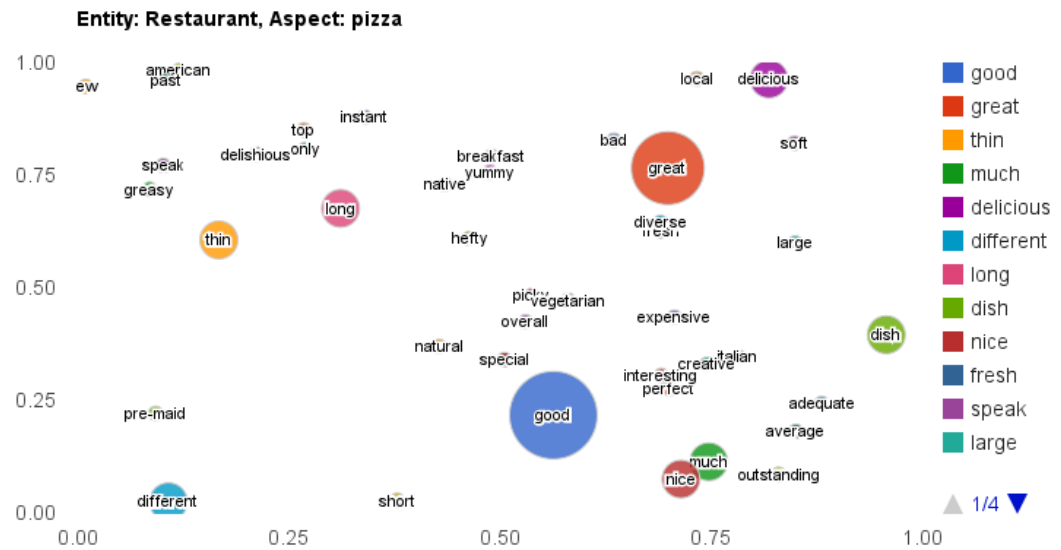
Figure 6.8: Screen shot of the page showing positive opinions (sentences) for the entity hotel.  
Source: Own elaboration.

Summary



See Opinions on this aspect... [GO!](#)

Summary



See Opinions on this aspect... [GO!](#)

Figure 6.9: Screen shots of the special pages for *aspect expressions*, showing adjective bubble charts for lake view for hotels (above) and for pizza for restaurants (below).

Source: Own elaboration.

# Chapter 7

## Conclusions

This work proposes an extension of the Liu’s aspect-based opinion mining technique in order to apply it to the tourism domain. The proposed extensions included the formalization of some ambiguous concepts, the development of new rules covering common cases that appear in the tourism domain, which are not encompassed by his models and the creation of special corpora or datasets for evaluation of the algorithms. Models to define and extract opinions from web documents present a simple, yet effective manner of transforming the unstructured data about opinions available on the web. In particular, the algorithm for *aspect expressions* extraction, based on frequent nouns and NPs appearing in reviews, achieved a poor performance in the tourism domain. Results show that multiple expressions are used to denote the same attribute or component of a tourism product (in this case, hotels and restaurants.) Therefore, not only the most frequent words need to be considered when extracting *aspect expressions* in order to achieve a better recall for this task. On the other hand, it was also discovered that users tend to *tell stories* when writing reviews about tourism products. This led to poor precision in the task of extracting *aspect expressions* since in reviews a lot of objects that are not components or attributes of the product are mentioned. All the extracted expressions that are not components or attributes of a product need to be filtered. In this context, the use of existing tourism ontologies or other methods of studying relations between words (as the one proposed by Popescu and Etzioni [48]) could be very useful.

Conversely, the application of Natural Processing Language rules for determining semantic orientation at the sentence level proved to be very effective for extracted *aspect expressions*, achieving an average precision and recall of 90%. Nevertheless, since *aspect expressions* that were extracted only represent a small percentage of the ones that were manually detected, the method needs to be tested for all possible expressions on the topic of tourism in order to give a more conclusive analysis. In particular, the main downside of the proposed rules seems to be the fact that they are not domain sensitive. Specific sentences regarding context or domain dependent topics need to be specially treated. In the tourism domain, this could represent a major problem since a lot of opinions could imply a positive or negative sentiment depending on the product the opinion is given on.

A different conclusion in relation to the performance evaluation of the proposed opinion algorithms indicates that the process of elaborating a linguistic corpus of opinions could easily become a very complex task. Nevertheless, through the participation of a linguistics expert in the annotation process, it was possible to more accurately understand how opinions are given by users and how opinion linguistic corpora should be elaborated. Documenting any corpora with all the assumptions, rules, techniques or methodologies that were used when generating the input texts or annotating is a key factor to a better understanding for those who may use those corpora. This was a main downside found in Liu's case, considering that in the opinions domain any annotation process will always be a somewhat subjective task.

On the other hand, the proposed software architecture, designed under the modular programming paradigm, permitted building a comprehensive background to support all the tasks needed when doing web opinion mining. Although Python was used here to implement the proposed requirements of the application, the abstraction of their proposals permits the use of almost any other programming language.

In particular, the architecture included a logic data model that supports an aspect-based opinion mining process. Since the model is built using the standard E-R notation, it can be included into any other relational data model and complemented with any other additional information.

Also, the use of Python and the NLTK proved how simple, yet powerful these tools are for NLP tasks. They proved that the implementation of the complex rules needed to extract *aspect expressions* or determine semantic orientation can be achieved by any user with basic or no experience in NLP. The use of the Django Framework for the Results Visualization Module permitted a very fast and result-oriented development of a web-based application for showing opinion mining results. This again proved that Python offers a simple, yet effective set of tools for programmers with little experience.

In conclusion, this work has proven that tourism product reviews available on web sites contain valuable information about customer preferences, conceived as individuals' attitudes toward a set of objects or evaluative judgments in the sense of liking or disliking these objects. In particular, it has also been proved that these preferences can be obtained by applying aspect-based opinion mining algorithms to hotel and restaurant tourism reviews. Indeed, it was also shown that some advanced techniques are even capable of measuring these preferences using aspect-based opinion mining results together with probabilistic or econometric models.

In addition, although the process that the opinion mining algorithms implement can be performed by human annotators, an automatic opinion mining system like the application that has been designed and developed here has two important advantages:

1. The system only takes one or two minutes to process 1000 reviews and extract all the opinions on them. On the other hand, as experienced when elaborating the annotated corpora, when it is performed by humans, the process of extracting opinions could take a significant amount of time, namely several hours or even days, considering typical



working time.

2. Secondly, and perhaps more importantly, the system offers consistency when extracting *aspect expressions* and when determining orientation of opinions. In other words, under the same parameters, the model will always return the same results. Conversely, when humans perform these kinds of tasks, inconsistencies and subjectivity are unavoidable, and the results of the process might depend on the specific situation or circumstance where it is carried out.

This study proposed a web opinion mining application whose objective was to permit users to gain insights about tourism product preferences in the X Región de Los Lagos. As shown, the application is able to provide users a platform to visualize and explore results from opinion mining algorithms applied on hotel and restaurant reviews obtained about the Lake District on TripAdvisor, preparing aspect-based summaries from the opinions retrieved and offering a set of other related features. The possible uses of an application like this are huge. However, in the context of the WHALE project, stakeholders are benefitted in specific ways.

- To clients and future customers, the system offers a simple manner of knowing what other customers think about hotels and restaurants in Los Lagos and discover those *aspects* that these products or services provide. With this information, users can make better decisions and choose a product or service that better satisfies them.
- To providers of tourism products in Los Lagos, the system offers a manner of measuring the customer satisfaction level, letting them know those *aspects* of their restaurants or hotels that customers liked or disliked the most and covering an important number of participants from different locations. In simple words, with the system, providers will have a better understanding of what their customers really want or need.

## 7.1 Future Work

The extended aspect-based opinion mining process presented here

1. Transform the implemented system into a fully object-oriented and modular platform. In this manner, different or new alternatives for each step in the process can be simply implemented and changed. To transform the application into an enterprise system or into a web service, seem also valid and interesting alternatives.
2. Consider more opinion or review sources and more tourism products. Particularly, in relation to the project, opinions given on the web site *patagonialoslagos* need to be considered in the future.
3. In order to improve the performance of the *aspect expressions* extraction task, ways of extracting infrequent and implicit *aspect expressions* need to be implemented. This also ensures that the information that is being delivered to users is complete and not partial.
4. Transforming *aspect expressions* into *aspects* represents another important challenge

that should be embraced in the future. Presenting *aspect expressions* to users implies redundancy and makes analysis more complex. By clustering or grouping *aspect expressions* into *aspects* the system becomes more intuitive and robust.

5. As seen in Chapter 6, tourism product reviews contain an important number of sentences that have no opinions. These sentences need to be filtered since they introduce noise to the opinion mining process.
6. Also, context- and domain-dependent opinions need to be considered. This ensures that the sentiment orientation is correctly determined for each opinion and that only true information is being shown to the system users.
7. On the other hand, since this work has proposed an extension of Liu's ideas, evaluating how this extension performs using the corpora provided by him is also proposed for future work. This should be performed in order to make advantages of the developed technique clearer.
8. Finally, a natural extension of this work implies working with opinionated documents, i.e. products reviews, written in Spanish (and other languages.) As seen, sentence and word tokenizers are usually sets of rules, or special machine learning algorithms that needs to be properly trained in order to generate good results. For Spanish, the main difficulties are either developing special rules to tokenize words and sentences, or finding high quality and long-enough corpora to train the algorithms. The same situation goes for POS-taggers and syntactical chunkers, which are the core of the *aspect expression* extraction phase. Some of the orientation determination rules also need to be adapted. On the other hand, data sources used in this work, namely TripAdvisor, seems to be a valid and useful source of product reviews in Spanish. However, the amount of reviews available is lower. In conclusion, it is probable that even using the adequate tools and collecting a considerable number of reviews, the performance of the algorithms will decrease for Spanish. However, only by experimentation it is possible to give more concluding results.

# Appendix

- A Paper: *“Identifying Customer Preferences about Tourism Products using an Aspect-Based Opinion Mining Approach”*

See next page.



17<sup>th</sup> International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

## Identifying Customer Preferences about Tourism Products using an Aspect-Based Opinion Mining Approach

Edison Marrese-Taylor<sup>a</sup>, Juan D. Velásquez<sup>a</sup>, Felipe Bravo-Marquez<sup>b</sup>, Yutaka Matsuo<sup>c</sup>

<sup>a</sup>Department of Industrial Engineering, University of Chile, República #701, Santiago, Chile

<sup>b</sup>Department of Computer Science, University of Chile, Blanco Encalada #2120, Santiago, Chile

<sup>c</sup>Institute of Engineering Innovation, Graduate School of Engineering, The University of Tokyo 2-11-16 Yayoi, Bunkyo-ku, Tokyo 113-8656, Japan

---

### Abstract

In this work we extend Bing Liu's aspect-based opinion mining technique to apply it to the tourism domain. Using this extension, we also offer an approach in considering a new alternative to discover consumer preferences about tourism products, particularly hotels and restaurants, using opinions available on the Web in the manner of reviews. An experiment is also conducted, using hotel and restaurant reviews obtained from TripAdvisor, to evaluate our proposals. Results showed that tourism product reviews available on web sites contain valuable information about customer preferences that can be extracted using an aspect-based opinion mining approach. The proposed approach proved to be very effective in determining the sentiment orientation of opinions, achieving a precision and recall of 90%. However, on average, the algorithms were only capable of extracting 35% of the explicit *aspect expressions*.

© 2013 The Authors. Published by Elsevier B.V.

Selection and peer-review under responsibility of KES International.

*Keywords:* opinion mining, aspect-based, tourism, customer preferences, natural language processing, web mining

---

### 1. Introduction

With the inception of the Web 2.0 and the explosive growth of social networks, enterprises and persons are increasingly using the content in these media to make better decisions [1], [2], [3]. More people check the opinions of other shoppers before buying a product when trying to make a good choice. On the other hand, for organizations, the vast amount of information available in a public manner on the Web could make polls, focus groups and some similar techniques an unnecessary requirement in market research.

In particular, results provided by aspect-based opinion mining techniques could represent a real alternative in finding customer preferences about a product. An aspect-based opinion mining approach permits analyzing opinions about product features, namely, product components and attributes. As established in Lancaster's *new theory of consumer demand*, customer preferences about a product are intrinsically related to its features, i.e. *aspects*, stating that consumer behavior is a process of choosing bundles of product characteristics or attributes inherent in goods and services, rather than simply choosing bundles of goods or services themselves [4]. Thus,

---

E-mail address: [emarrese@ing.uchile.cl](mailto:emarrese@ing.uchile.cl).

discovering what these features are and defining how customers feel about these features may undoubtedly lead to a better comprehension of consumer preferences.

The aspect-based opinion mining is widely known and used in scientific research, but an extensive bibliographic revision has shown that the work of Liu et al. is probably the most complete one in this context. This led us to take his ideas as inspiration. Nevertheless, some original definitions and assumptions were analyzed and modified to fit this problem context. Thus, in this work we propose an extension of the Liu's aspect-based opinion mining methodology in order to apply it to the tourism domain. The proposed extensions will include the formalization of some ambiguous concepts, the development of new rules covering common cases that appear in the tourism domain and are not encompassed by his models (like *aspects* appearing more than once in a sentence) and the creation of special corpora or datasets for evaluation of the algorithms.

We state that a system implementing the extended aspect-based opinion mining techniques we present would permit its users to discover customer preferences when applied to tourism product reviews. These preferences are conceived, as introduced in scientific research by psychology, as an individual's attitude towards a set of objects, typically reflected in an explicit decision-making process [5] or as an evaluative judgment in the sense of liking or disliking an object [6]. For instance, in the case of the tourism industry, the study of customer preferences is usually implemented using traditional tools, which, in general, fail to cover a significantly representative number of participants because they are applied to specific groups of people. In this context, aspect-based opinion mining offers a set of techniques to deal with data about opinions and deliver an alternative and large scope method to understand consumer preferences.

The rest of this paper is structured in the following manner. In the first place, we briefly present most important ideas about aspect-based opinion mining in section 2, with a special mention to the approach developed by Liu et al. After that, in section 3, we extend Liu's ideas and propose our own approach. Later, in section 4, we present an experimental setup to evaluate how the proposed approach performs for tourism product reviews in the X Región de Los Lagos, Chile, also showing most important results. This intends to encompass the current situation in the region, where tourism operators try to understand customer preferences using studies with limited scope. Finally, in section 5, main conclusions of this paper and future work are detailed.

## 2. Previous Work

Aspect-based opinion mining techniques divide input texts into *aspects*, also called features or subtopics in literature, that usually correspond to arbitrary topics considered important or representative of the text that is being analyzed. The aspect-based approach is very popular and many authors have developed their own perspectives and models. Examples of this are the works of Lu et al. [7], Popescu and Etzioni [8], Archak et al. [9], Decker and Trusov [10], Ku et al. [11], Titov and McDonald [12], Zhuang et al. [13] and Zhao and Li [14]. However, research showed that the work of Liu et al. is probably the most comprehensive one in this context and that is why it was used here by us as inspiration. Comprehensive revisions of the state-of-the-art opinion mining techniques can be found in [15], [16] and [17].

### 2.1. Initial Definitions

In case of Liu's approach, as taken from [16], he proposes that opinions are 5-tuples, composed of (1) An *entity*: Proposed to denote the opinion objective or, in other words, what is being evaluated by the opinion. An *entity* can contain a set of components and attributes and, similarly, each *entity* component can have its own subcomponents and attributes. Finally, an *entity* can be decomposed into a tree or hierarchy of subattributes and subcomponents. (2) An *aspect*: Because it is difficult to study an *entity* at an arbitrary hierarchy level, this hierarchy is simplified to one or two levels, denoting *aspect* every component or attribute of the *entity*. In this way, the root of the hierarchy or tree becomes the *entity* itself, each leaf is an *aspect* and links are part-of relations. (3) The *Sentiment orientation*, considering that opinions express a *positive* or *negative* sentiment about what they evaluate. (4) The *Opinion holder*, which corresponds to the user (a person, an enterprise, etc.) that gives the opinion. (5) *Time*: Time and date when the opinion was given. Then, in few words, opinions are considered to be a positive or negative view, attitude, emotion or appraisal about an entity or an aspect of that entity from an opinion holder in a specific time. The following concepts are also introduced:

- *Entity expression*: Corresponds to the actual word or phrase written by the user to denote or indicate an *entity*. As a result, *entities* are then generalizations of every *entity expression* used in the analyzed documents, or a particular realization of an entity expression. In [16] this concept is called *entity name*.
- *Aspect expression*: As for an *entity expression*, the *aspect expression* is the actual word or phrase written by the user to denote or indicate an *aspect*. Thus, aspects are also general concepts that comprise every *aspect expression*. They are called *aspect names* by Bing Liu.

It is then possible to define a model of an *entity* and a model of an opinionated document. In this manner, an *entity*  $e_i$  is represented by itself as a whole and a finite set of *aspects*,  $A_i = \{a_{i1}, a_{i2}, \dots, a_{in}\}$ . The *entity* can be expressed with any one of a final set of *entity expressions*  $EE_i = \{ee_{i1}, ee_{i2}, \dots, ee_{is}\}$ . Each *aspect*  $a_{ij}$  of  $A_i$  of *entity*  $e_i$  can be expressed by any one of a finite set of *aspect expressions*  $AE_{ij} = \{ae_{ij1}, ae_{ij2}, \dots, ae_{ijm}\}$ . On the other hand, an opinionated document  $d_k \in D$  contains opinions on a set of *entities*  $e_1, e_2, \dots, e_r$  from a set of opinion holders  $h_1, h_2, \dots, h_p$ . The opinions on each *entity*  $e_i$  are expressed on the *entity* itself and a subset  $A_{ik}$  of its *aspects*.

## 2.2. Process Steps

Kim et al. give a good review of historical and state-of-the-art aspect-based developments in [15]. The authors indicate that the process is commonly made up of three distinct steps, which are also considered by Liu.

1. Aspect identification, to find and extract important topics in the text that will then be used to summarize. In [18] Hu and Liu present a technique based in NLP and statistics. In that case, part-of-speech (POS) tagging and syntax tree parsing (chunking) are used to find nouns and noun phrases or NPs. Then, using frequent itemset mining, the most frequent nouns and NPs are extracted and became aspect candidates. Finally, some special linguistic rules are implemented in order to assure that the terms inside aspects composed of more than one word are likely to represent a coherent object together, also eliminating redundant aspects. This proposal, although fairly effective and complete, lacked to define some key concepts that remained somewhat ambiguous. Since our proposal considers Liu's approach to extract *aspects*, all these ambiguous concepts will be defined by us in next section.
2. Sentiment Prediction, to determine the sentiment orientation on each aspect. Ding, Liu and Yu offer a lexicon and rule based approach in [19]. This method relies on a sentiment word dictionary that contains a list of positive and negative words (called *opinion words*) which are used to match terms in the opinionated text. Also, since other special words appearing in the text and not included in the lexicon might also change the orientation, special linguistic rules are proposed. For instance, negation words, like *no* or *not*, change the opinion orientation. Nevertheless, despite how simple these rules might appear, it is important to handle them with care, because not all occurrences of such rules or word apparitions will always mean the same. In this context, the rules developed by Ding, Liu and Yu in are probably the most comprehensive ones. This rules also include an aggregation score function to determine the orientation of an aspect in a sentence combining multiple *opinion words*. This function will be explained in detail in next section, since it will be used and extended by us to cover some special cases that commonly appear in tourism product reviews available on the web.
3. Summary Generation, to present processed results in a simple manner. In this context, opinion quintuples defined are a good source of information for generating quantitative summaries. In particular, Liu defines a kind of summary called aspect-based opinion summary [20] [21], that consists of bar charts that show the number of positive and negative opinions about every *aspect* of one *entity*. In [22], Liu also proposes that the bar charts could be used to compare a set of selected products, showing the set of all aspects of the chosen products in the chart. In this case, each bar above or below the x-axis can be displayed in two scales: (1) The actual number of positive or negative opinions normalized with the maximal number of opinions on any feature of any product and (2) The percent of positive or negative opinions, showing the comparison in terms of percentages of positive and negative reviews.

### 3. Proposed Extension

As seen, the discovery and construction of the opinion tuples is achieved by following a set of structured steps. In the following sections, discovered issues on each one of the tree steps of the aspect-based opinion mining process as defined by Liu are discussed, also explaining solutions we propose on each case. All these inconsistencies came to light when the model was applied to real data about tourism product reviews.

#### 3.1. Aspect expression extraction

As defined by Liu, *aspects* do not directly appear in a text but they exist in the manner of *aspect expressions*. Then, when trying to apply Liu's opinion model to extract opinions from real data, concepts are somewhat confusing or unclear. On the other hand, it is also unclear how *aspects* that appear more than once in a document are managed in building tuples: the model fails to clearly specify the text granularity to consider when constructing tuples from an opinionated document. Having noticed these issues, a model to build opinion tuples from an opinionated document has been developed here.

To make things simpler, consider a set of opinionated documents  $D_i = \{d_{i1}, d_{i2}, \dots, d_{im}\}$  about only one *entity*,  $e_i$ . This seems a realistic assumption since opinions are usually disposed in the manner of product reviews on the web. Then, each opinionated document will correspond to a review or opinion given by holder  $h_k$  in time  $t_k$ . Let  $S_{ik}$  be the set of all sentences in  $d_{ik}$ , with  $S_{ik} = \{s_{ij1}, s_{ij2}, \dots, s_{ijm}\}$ . Opinions on  $e_i$  in  $d_{ik}$  will be expressed on the *entity* itself and on a subset  $A_{ik}$  of its *aspects*. Similarly, each *aspect* of  $A_{ik}$  will appear on  $d_{ik}$  as a set of *aspect expressions*  $AE_{ijk}$ , subset of  $AE_{ij}$ . The entity  $e_i$  will appear as a subset of different *entity expressions*  $EE_{ik} \subseteq EE_i$ . Thus, the set  $EX_{D_i}$  is defined as the set of all *aspect expressions* of all *aspects* and all *entity expressions* appearing in  $D_i$ . A sentence is related to one *aspect expression* or *entity expression* only if it appears in that sentence. Then, sentiment orientation needs to be determined for each pair  $(ex, s)$  only if any *aspect expression* or *entity expression* appears on it. After determining sentiment orientation,  $h_k$  and  $t_k$  of the corresponding document  $d_{ik}$  should simply be added in order to build each opinion tuple.

On the other hand, Liu's proposal indicates that it seems reasonable that frequently talked nouns in product reviews are usually genuine and important *aspects expressions* because when people comment on different *aspects* of a product, the vocabulary that they use usually converges. Nevertheless, two main reasons explain the fact that many different expressions could indicate the same concept:

- The economy principle in languages [23] indicates that they try to say a lot using few words. For example, the sentence “*The hotel has good wifi.*” corresponds to a lexicalization, where the original expression, “*The hotel has good internet access through wifi.*”, is shortened according to the economy principle.
- Each language presents systems that organize its concepts, also pursuing simplification. For that reason, many words in English (as in all other languages) simply are hyponyms of a determined hypernym. An hyponym is a word or phrase whose semantic field is included within that of another word, its hypernym. For instances, scarlet, vermilion, carmine, and crimson are all hyponyms of red (their hypernym), which is, in turn, a hyponym of color [24].

Then, in practice, finding the *aspects* that are evaluated in a set of opinionated documents is a really complex task. In fact, detecting *aspect expressions* from a set of documents with opinions should be a completely different task than defining or finding the real *aspects* in them, because the amount of possible expressions appearing in a text is really huge. In the tourism domain, experiments showed (see section 4), that several expressions are in fact used to refer to the same concept, which are then not extracted by the frequency-based algorithm that Hu and Liu propose in [18].

A different issue found in Liu's proposals is related to the concepts of sentence and *word distance*, that although widely used, are not clearly defined in his work. Despite deeper linguistic analysis, here we will define a sentence to be an ordered set of tokens, including words and punctuation. One token that appears in two different positions must be considered twice, as the positions where they appear are distinct. In other words, a sentence  $S$  will correspond to a set of unique tuples  $(token, position)$ . Positions can only be in  $\mathbb{N} \cup \{0\}$  and the difference between

two adjacent components must be 1. Then, the concept of *word distance* between two elements of sentence  $S$  will correspond to the difference of the positions of the two tokens in  $S$ .

$$\text{Word Distance}(t_a, t_b) = |\text{position}(t_a) - \text{position}(t_b)| \text{ with } t_a, t_b \in S \quad (1)$$

As  $\text{Word Distance}(t_a, t_b)$  is simply the absolute value of the difference between numbers in  $\mathbb{N} \cup \{0\}$ ,  $\text{Word Distance}(t_a, t_b)$  is a metric on the set  $S$  as it satisfies the conditions of non-negativity, identity of indiscernibles, symmetry and triangle inequality. Note that minimal distance between 2 elements in  $S$  is 1, and it occurs between adjacent elements. The maximum distance corresponds to  $|S| + 1$ . Considering these definitions, we apply the technique developed by Hu and Liu in [18].

### 3.2. Determination of the Opinion Orientation

Taking Liu's work in [19] as inspiration, a set of rules to determine the sentence orientation was developed, always considering *opinion words* as a basis.

#### 3.2.1. Word Orientation Rules

- **Word Rules:** Positive *opinion words* will intrinsically have a *score* of 1, denoting a normalized positive orientation, while negative ones will have associated a *score* of  $-1$ . Every noun and adjective in each sentence that is not an *opinion word* will have an intrinsic *score* of 0 and will be called *neutral word*.
- **Negation Rules:** A negation word or phrase usually reverses the opinion expressed in a sentence. Consequently, *opinion words* or *neutral words* that are affected by negations need to be specially treated. Three rules must be applied: Negation Negative  $\rightarrow$  Positive, Negation Positive  $\rightarrow$  Negative and Negation Neutral  $\rightarrow$  Negative. Negation words and phrases include: *no, not, never, n't, dont, cant, didnt, wouldnt, havent, shouldnt* (misspellings are here intentional). Also, some negation patterns are considered, including *stop + vb-ing, quit + vb-ing* and *cease + to + vb*.
- **Too Rules:** Sentences where words *too, excessively* or *overly* appear, are also handled specially. When an *opinion word* or a *neutral word* appears near one of the mentioned terms, denoted *too words*, its orientation will always be Negative (*score* =  $-1$ ).

#### 3.2.2. Aspect Orientation Rules

Having mentioned rules that help in determining each word orientation in a sentence, it is now explained how all these orientations should be combined to determine the final orientation of a sentence on a particular aspect. This algorithm should only consider words marked as *opinion words* or *neutral words* as they are the only ones that will provide an orientation for each sentence.

- **Aspect Words Aggregation Rule:** Let  $s$  be a sentence that contains the set of *aspect expressions*  $A = \{a_1, \dots, a_m\}$ , each one of them appearing only one time in  $s$ . Also, let  $AW_i$  be the set of words that compose aspect  $a_i$ , where  $AW_i = \{aw_{i1}, aw_{i2}, \dots, aw_{in}\}$ . Each  $aw_{ij}$  will be called *aspect word* and it will correspond to *aspect expression*  $a_i$ . If *scores* for each *opinion word* and *neutral word* in  $s$  are known, *score* for each  $aw_{ij}$  in  $s$  is given by the following aggregation function:

$$\text{score}(aw_{ij}, s) = \sum_{ow_j \in s} \frac{\text{score}(ow_j)}{\text{Word Distance}(ow_j, aw_{ij})} \quad (2)$$

Where  $ow_j$  is an *opinion word* or *neutral word* in  $s$ ,  $\text{Word Distance}(ow_j, aw_{ij})$  is the word distance between the aspect word  $aw_{ij}$  and the opinion word  $ow_j$  in  $s$ . This function we take from Ding, Liu and Yu's work. Nevertheless, although used by them, their proposition lacked to explain how should it be applied to *aspects* that are composed of more than one word (are *compound*), only considering one position for each *aspect*. In the tourism domain, some important aspects are *compound*. For instance, in the sentence "*The hotel had a poor view of the beautiful lake.*" an *aspect* that is extracted by Liu is *lake view*. However, Liu's proposal does not explain how the orientation on this *aspect* in the sentence should be obtained. Accounting on cases like this, here we propose that orientation should be obtained for each word in each *aspect*. Orientations are aggregated according to next rule.



- **Aspect Aggregation Rule:** For each *compound aspect expression*  $a_i$  in  $s$ , its orientation will be calculated considering the *scores* of all the words that compose it,  $aw_{ij} \in A_i$ , according to the following equation.

$$score(a_i, s) = \sum_{j \in A_i} score(aw_{ij}, s) \quad (3)$$

- **Position Aggregation Rule:** Another case that is not covered by Liu’s proposals, that commonly appears in tourism product reviews, is that *aspect expressions* could appear more than once in a sentence. For instance, a sentence extracted from a hotel review says: “When we arrived to the hotel it looked really good and only after trying several rooms we discovered the whole hotel was really mouldy in the interior.” Clearly, a method for determining the final opinion orientation on *aspect expressions* in cases like this is needed. Supposing that  $a_i$  appears  $t$  times in  $s$  and knowing the score of each *aspect expression* appearance  $a_i^k, k \in \{1, 2, \dots, t\}$ , we propose that the final score of  $a_i$ , or  $fscore(a_i, s)$ , should be calculated by simply adding the values of the scores of all the  $a_i$  appearances in  $s$ , according to the following equation.

$$fscore(a_i, s) = \sum_{k=1}^t score(a_i^k, s) \quad (4)$$

Note that when  $a_i$  only appears one time in  $s$ ,  $fscore(a_i, s) = score(a_i, s)$ . Then, for each *aspect expression*, if  $fscore(a_i, s)$  is positive, the opinion is considered positive on  $a_i$  and if it is negative, the opinion is considered negative on  $a_i$ . If none of these cases occur, the opinion orientation is zero.

- **But Clauses Rules:** In this case, we use exactly the same rule that Liu proposes in [19]. This rule states that when a *but word*  $b$  appears in sentence  $s$ ,  $s$  must be broken into two segments, the one before and the one after  $b$ . If the orientation of any *aspect word*  $aw_{ij}$  appearing in the sentence segment after  $b$  is zero, its orientation should then be determined using the segment before  $b$  and negating it. We realized that a little ambiguity existed since in some of these cases  $aw_{ij}$  may not truly be inside the segment that is considered to determine the orientation of  $s$ . Here, we simply propose that  $aw_{ij}$  must be added at the final position of the corresponding segment in order to avoid the consistency issue.

### 3.3. Summarization

Liu’s proposal seems fairly simple and effective for summarizing opinions. However, it lacks a robust way of measuring the importance of each evaluated *aspect*. In [20], *aspects* are ranked according to the frequency of their appearances in the reviews, but it is also declared that other types of rankings are also possible, like ranking *aspects* according to the number of reviews that express positive or negative opinions. Here, a new proposal was developed, attempting to measure the importance of each *aspect* simultaneously using the amount of positive and negative opinions of it. The underlying assumption is that an *aspect* that has a lot of positive and negative opinions will be more important, since the high number of opinions of both orientations might indicate that customers are very interested in that *aspect*. In this way, not only the total number of times that an *aspect* appears is considered in measuring importance, but also the dispersion in the number of positive and negative opinions. Let  $P_i$  and  $N_i$  be the number of positive and negative opinions on aspect  $a_i, i \in \{1, \dots, n\}$ . Then,  $PScore_i$  and  $NScore_i$  will be the min-max normalized values of  $P_i$  and  $N_i$ , respectively. With this, we calculate the standard deviation of these *Scores* using:

$$STDScore_i = \sqrt{\frac{1}{2} \left( \left( PScore_i - \frac{PScore_i + NScore_i}{2} \right)^2 + \left( NScore_i - \frac{PScore_i + NScore_i}{2} \right)^2 \right)} \quad (5)$$

Thus, we define our new measure for each *aspect*  $a_i$ , called *Relative Importance*, as the min-max normalized value of its  $STDScore_i$ . We propose that aspect-based summaries should include bar charts and a table that shows the actual values of  $PScore_i, NScore_i$  and *Relative Importance* for each *aspect expression*.

#### 4. Experiments and analysis

The experiment we carried out consisted in evaluating how the proposed opinion mining algorithms perform when applied to tourism product reviews from Los Lagos, particularly, hotels and restaurants. Our work here mainly consisted in: (1) Generating annotated corpora or datasets to evaluate the performance of the algorithms for the selected products, using the site TripAdvisor as a source, (2) Measuring the performance of the algorithms and (3) Designing and developing an application that permit users to extract opinions from these reviews and see proposed summarization charts. This application was implemented using Python, the Natural Language Toolkit (NLTK) for NLP tasks and the Django Framework, and included modules that helped carrying out (1) and (2).

##### 4.1. Annotated Corpora

Using a web crawler module included in the the application, we downloaded reviews of hotels and restaurants in *Lake District* from TripAdvisor. The downloaded reviews of each product were saved in a CSV file, which we used to randomly select, on each case, 100 reviews that were used to build the annotated corpora. The annotation process followed the spirit of what Liu proposes in [20] and [19]: each review was tokenized into sentences using [25] and was manually annotated following a set of rules and a notation system that had previously been designed by us (for details see our corpora material that will be available upon acceptance.) However, sentences that seemed ambiguous or really difficult to tag were discussed with a second human annotator, an expert in linguistics. Once an agreement was achieved, the sentence was tagged according to that agreement. This marks an important difference between this study and other tagging procedures commonly carried out in literature, where different annotators tag the same corpus separately and only once the annotation procedure has finished are different results of the same corpus compared to define the final choice. This different approach was used here due to time constrains, since it seemed more efficient and was worth trying as a contribution to research in this field.

Table 1 gives general a description of the generated corpora. In both cases, almost 80% of the sentences contained opinions. This shows that opinionated sentences represent an important fraction of the total sentences, which somehow validates the use of TripAdvisor as a source of opinions for hotels and restaurants. Nevertheless, non-opinionated sentences are also a considerable number, consequently introducing noise into the opinion-extraction process.

Feature	Hotels Corpus	Restaurants Corpus
Number of Reviews	100	100
Total Number of Sentences	789	470
Number of Opinion Sentences	609	368
Opinion Sentences/Sentences	77.19%	78.3%

Table 1. Corpora Details.

Table 2 gives detail about the *aspect expressions* that were manually extracted. Following Liu's notation, we call explicit *aspect expressions* to those expressions that appear in the manner of nouns or NPs in a sentence and implicit *aspect expressions* to all other cases. Results show that in both corpora, explicit *aspect expressions* are the most common ones, representing around 70% of of all the extracted expressions. Also, note that some *aspects expressions* appear in both an explicit and implicit manner. Since they are explicit expressions in at least one case, they were considered as explicit *aspect expressions*. On the other hand, extracted *aspect expressions* that are purely implicit are also an important number, being almost 20% in both cases. A simple review showed that most of these aspects were indicated by adjectives.

Aspect Expressions	Hotels Corpus		Restaurants Corpus	
	Number	Percentage	Number	Percentage
Explicit	229	73.87%	161	67.93%
Explicit and Implicit	30	9.68%	26	10.97%
Implicit	51	16.45%	50	21.1%
Total	310	100%	237	100%

Table 2. Detail on *aspect expressions* found in corpora.

#### 4.2. Algorithms performance

To measure the performance of the algorithms, three tasks will be evaluated by comparing its results with the manually annotated corpora: (1) Explicit aspect extraction, measuring the effectiveness of explicit *aspect expression* extraction, (2) Subjectivity classification, to evaluate the effectiveness of opinion sentence extraction and (3) Sentiment classification, to measure the accuracy of orientation prediction of each pair (*ex, s*) (*aspect expression, sentence*) for the positive class. Here we present the best general performance obtained by doing a sensitivity analysis regarding the most sensitive parameter - the minimum support rule to extract *aspect expressions*, as defined in [18]. Precision, recall and f-measure were calculated for six different values of this parameter for each task. Then, best model was chosen using f-measure. Table 3 shows the obtained values.

Corpus	Hotels Corpus		Restaurants Corpus		Average Performance		
Index Name	Precision	Recall	Precision	Recall	Precision	Recall	F-measure
Explicit Aspect Extraction	33%	29%	42%	37%	38%	33%	36%
Subjectivity Classification	79%	93%	81%	88%	80%	<b>91%</b>	85%
Sentiment Classification	89%	93%	91%	93%	<b>90%</b>	<b>93%</b>	<b>92%</b>

Table 3. Performance results.

These results show that performance on the *aspect extraction* task is fairly poor in the tourism domain. The algorithm is only capable of extracting almost 30% of the total explicit expressions for hotels and almost 40% for restaurants. Moreover, a high percentage of the extracted expressions do not correspond to real *aspect expressions* for both cases. The results tell two important things:

- When users write comments about hotels or restaurants, they mention many objects (i.e. nouns or NPs) that do not correspond to attributes or components of the hotel or restaurant. In other words, users probably *tell stories* about their experiences when writing reviews about tourism products. This may explain the low precision obtained for the explicit *aspect extraction* task in both cases. For instance, in the case of hotels, users commonly refer to objects like time, day and city, which, although relevant for stories, tell nothing about the hotel.
- As stated before, users tend to use many different expressions to refer to the same component or attribute of a hotel or restaurant. In this manner, since many different *aspect expressions* are used, most of them will appear in low frequencies in the text. Consequently, itemsets that do not occur with relative high frequencies need to be considered in order to be extracted. In [18], Liu proposed a method to extract these infrequent *aspect expressions* by exploiting their relationships with frequent *opinion words*. In this study, this method was not considered since in Liu's case, the extracted infrequent *aspect expressions* only represented an improvement of 15% for recall, at the cost of decreasing precision in almost 7%. However, given the poor results that have been obtained, it seems interesting to evaluate how this step improves or worsens performance in this case.

On the other hand, As Liu states in [18] and as seen in previous sections, the reason that probably explains precision being a little lower than recall in the task of subjectivity classification is the fact that users like to tell stories about experiences they lived in relation to products, in our case, hotels or restaurants. In these stories, there are some sentences that do not contain opinions. Since the application labels some of these sentences as opinion sentences because they contain both product *aspect expressions* and some *opinion words*, precision decreases. Nevertheless, although these sentences may not show strong user opinions towards the product features, they may still be beneficial and useful [18].

Finally, sentiment classification shows fairly good results. However, in this case most of the possible conclusions are difficult to prove because this task was only evaluated for those *aspect expressions* that were extracted by the application. Since these are somewhat the simplest ones, determining the sentiment orientation on them may be easier. Consequently, precision and recall could decrease when all *aspect expressions* are considered.

#### 4.3. Summary Visualization

The application we built permit users to see aspect-based summaries as proposed in section 3. Besides bar charts for each *entity* in the system, table shows the actual values of the *Positive Score*, *Negative Score* and *Relative*

Importance for each *aspect expression*; figure 1 shows an example. By clicking the name of each column, the table and the bar chart are sorted according to the clicked column (each click alternates between an ascending or descending sort.) Also, below the chart and the table, a list with all the *aspect expressions* is shown. By clicking one them, the user is redirected to a page showing specific information on that *aspect expression*. These special pages include adjective bubble chart that are built using the 2 nearest adjectives to each *aspect expression* in all

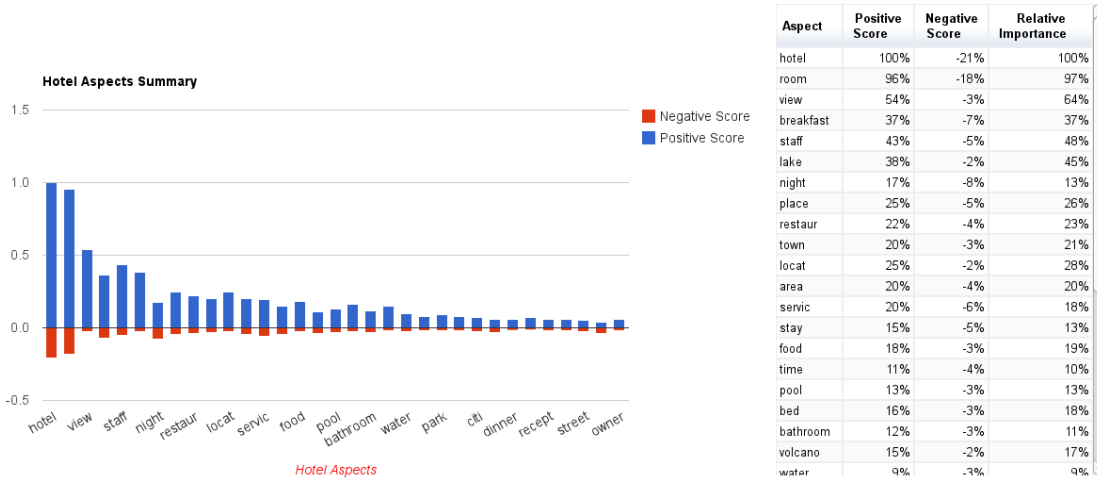


Figure 1. Proposed aspect-based summary for the entity *hotel* in lake District, using opinions extracted from hotels reviews in Lake District (Chile) in TripAdvisor. In the chart, *aspect expression* bars are ordered according to *Relative Importance* in a descending manner.

### 5. Conclusion

In the first place, the proposed models to define and extract opinions from web documents present a simple, yet relatively effective manner of transforming the unstructured data about opinions available on the web. In particular, the algorithm for *aspect expressions* extraction, based on frequents nouns and NPs appearing in reviews, achieved a poor performance in the tourism domain. Results show that multiple expressions are used to denote the same attribute or component of a tourism product (in this case, hotels and restaurants.) Therefore, not only the most frequent words need to be considered when extracting *aspect expressions* in order to achieve a better recall for this task. On the other hand, it was also discovered that users tend to *tell stories* when writing reviews about tourism products. This led to poor precision in the task of extracting *aspect expressions* since in reviews a lot of objects that are not components or attributes of the product are mentioned. All the extracted expressions that are not components or attributes of a product need to be filtered. In this context, the use of ontologies, as in [26], [14] and [27], or other methods of studying relations between words (as the one proposed in [8] or in [28]) could be very useful. Conversely, the application of NLP rules for determining semantic orientation proved to be very effective for extracted *aspect expressions*, achieving an average precision and recall of 90%. Nevertheless, since *aspect expressions* that were extracted only represent a small percentage of the ones that were manually detected, the method needs to be tested for all possible expressions on the topic of tourism in order to give a more conclusive analysis; this is proposed for future work.

On the other hand, one important downside of the proposed rules seems to be the fact that they are not domain sensitive. Specific sentences regarding context or domain dependent topics need to be specially treated. In the tourism domain, this could represent a major problem since a lot of opinions could imply a positive or negative sentiment depending on the product the opinion is given on. A method of dealing with these issues, although proposed in [19], was here left for future work. Using different state-of-the-art-methods to determine the sentiment orientation could also solve this problem. On the other hand, considering that in tourism product reviews an important number of sentences do not contain opinions - which led to poor precision in the task of subjectivity classification, applying methods to filter these sentences seems crucial. Also, we realized that this task could easily become very complex. Nevertheless, through the participation of a linguistics expert in the annotation process, it

was possible to more accurately understand how opinions are given by users and how opinion linguistic corpora should be elaborated. Documenting any corpora with all the assumptions, rules, techniques or methodologies that were used when generating the input texts or annotating is a key factor to a better understanding for those who may use those corpora. This was a main downside found in Liu's case, considering that in the opinions domain any annotation process will always be a somewhat subjective task. Finally, since we have proposed an extension of Liu's proposals, we also intend to evaluate how our extension performs using the corpora provided by him, in order to make our proposal advantages clearer. This is also proposed for future work.

## Acknowledgements

This work was supported partially by the FONDEF project D10I-1198, entitled WHALE: *Web Hypermedia Analysis Latent Environment* and the Millennium Institute on Complex Engineering Systems (ICM: P-05-004-F, CONICYT: FBO16).

## References

- [1] D. Park, S. Kim, The effects of consumer knowledge on message processing of electronic word-of-mouth via online consumer reviews, *Electronic Commerce Research and Applications* 7 (4) (2009) 399–410.
- [2] H. Shin, D. Hanssens, K. Kim, B. Gajula, Impact of positive vs. negative e-sentiment on daily market value of high-tech products.
- [3] F. Zhu, X. Zhang, Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics, *Journal of Marketing* 74 (2) (2010) 133–148.
- [4] K. Lancaster, A new approach to consumer theory, *The journal of political economy* 74 (2) (1966) 132–157.
- [5] S. Lichtenstein, P. Slovic, *The construction of preference*, Cambridge University Press, 2006.
- [6] K. Scherer, What are emotions? and how can they be measured?, *Social science information* 44 (4) (2005) 695–729.
- [7] Y. Lu, C. Zhai, N. Sundaresan, Rated aspect summarization of short comments, in: *Proceedings of the 18th international conference on World wide web*, ACM, 2009, pp. 131–140.
- [8] A. Popescu, O. Etzioni, Extracting product features and opinions from reviews, in: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2005, pp. 339–346.
- [9] N. Archak, A. Ghose, P. Ipeirotis, Show me the money!: deriving the pricing power of product features by mining consumer reviews, in: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2007, pp. 56–65.
- [10] R. Decker, M. Trusov, Estimating aggregate consumer preferences from online product reviews, *International Journal of Research in Marketing* 27 (4) (2010) 293–307.
- [11] L. Ku, Y. Liang, H. Chen, Opinion extraction, summarization and tracking in news and blog corpora, in: *Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*, no. 2001, 2006.
- [12] I. Titov, R. McDonald, A joint model of text and aspect ratings for sentiment summarization, *Urbana* 51 (2008) 61801.
- [13] L. Zhuang, F. Jing, X.-Y. Zhu, L. Zhang, Movie review mining and summarization, in: *Conference on Information and Knowledge Management: Proceedings of the 15 th ACM international conference on Information and knowledge management*, Vol. 6, 2006, pp. 43–50.
- [14] L. Zhao, C. Li, Ontology based opinion mining for movie reviews, *Knowledge Science, Engineering and Management* (2009) 204–214.
- [15] H. Kim, K. Ganesan, P. Sondhi, C. Zhai, Comprehensive review of opinion summarization.
- [16] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, Springer Verlag, 2007.
- [17] B. Pang, L. Lee, Opinion mining and sentiment analysis, *Foundations and Trends in Information Retrieval* 2 (1-2) (2008) 1–135.
- [18] M. Hu, B. Liu, Mining opinion features in customer reviews, in: *Proceedings of the National Conference on Artificial Intelligence*, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004, pp. 755–760.
- [19] X. Ding, B. Liu, P. Yu, A holistic lexicon-based approach to opinion mining, in: *Proceedings of the international conference on Web search and web data mining*, ACM, 2008, pp. 231–240.
- [20] M. Hu, B. Liu, Mining and summarizing customer reviews, in: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2004, pp. 168–177.
- [21] M. Hu, B. Liu, Opinion extraction and summarization on the web, in: *Proceedings Of The National Conference On Artificial Intelligence*, Vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1621.
- [22] B. Liu, M. Hu, J. Cheng, Opinion observer: analyzing and comparing opinions on the web, in: *Proceedings of the 14th international conference on World Wide Web*, ACM, 2005, pp. 342–351.
- [23] A. Vicentini, The economy principle in language, *Notes and Observations from Early Modern English Grammars*. *Mots, Palabras, Words* 3 (2003) 37–57.
- [24] V. Fromkin, R. Rodman, N. Hyams, *An introduction to language*, Wadsworth Publishing Company, 2010.
- [25] T. Kiss, J. Strunk, Unsupervised multilingual sentence boundary detection, *Computational Linguistics* 32 (4) (2006) 485–525.
- [26] A. Cadilhac, F. Benamara, N. Aussenac-Gilles, Ontolexical resources for feature based opinion mining: a case-study, in: *23rd International Conference on Computational Linguistics*, 2010, p. 77.
- [27] E. Vallés Balaguer, P. Rosso, A. Locoro, V. Mascardi, Análisis de opiniones con ontologías, *Polibits* (41) (2010) 29–36.
- [28] D. Bollegala, Y. Matsuo, M. Ishizuka, An integrated approach to measuring semantic similarity between words using information available on the web, in: *HLT-NAACL*, 2007, pp. 340–347.

# References

- [1] IALE Tecnología S.L. e IALE Tecnología Chile Ltda., “Observatorio de inteligencia regional de turismo, como plataforma de observación, monitoreo e insumo para un desarrollo sustentable en la región de los lagos. informe final presentado al clúster de turismo de intereses especiales (abril de 2010),” *Santiago de Chile: IALE Tecnología*, 2010.
- [2] S. Pal, V. Talwar, and P. Mitra, “Web mining in soft computing framework: Relevance, state of the art and future directions,” *Neural Networks, IEEE Transactions on*, vol. 13, no. 5, pp. 1163–1177, 2002.
- [3] Juan D. Velásquez and V. Palade, “Adaptive web sites: A knowledge extraction from web data approach,” in *Proceeding of the 2008 conference on Adaptive Web Sites: A Knowledge Extraction from Web Data Approach*, pp. 1–272, IOS Press, 2008.
- [4] N. Zhong, J. Liu, Y. Yao, and S. Ohsuga, “Web intelligence (wi),” in *Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International*, pp. 469–470, IEEE, 2000.
- [5] Instituto Nacional de Estadística - INE, “Turismo, Informe Anual 2008,” 2008.
- [6] F. Molina, “Presentacion Final Proyecto Ballena, Puerto Montt,” 2011.
- [7] Juan D. Velásquez and P. Gonzalez, “Expanding the possibilities of deliberation: The use of data mining for strengthening democracy with an application to education reform,” *The Information Society*, vol. 26, no. 1, pp. 1–16, 2010.
- [8] W3C, “Word Wide Web Consortium (W3C) Web Site.” <http://www.w3.org/WWW/>, 2012. Seen on November 17th, 2012.
- [9] P. L. Heufemann, “Análisis del comportamiento del usuario en la web a partir de la simulación de su navegación usando colonia de hormiga,” Master’s thesis, Universidad de Chile, 2011.
- [10] F. Gorunescu, *Data Mining: Concepts, models and techniques*, vol. 12. Springer, 2011.
- [11] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus, “Knowledge discovery in databases: An overview,” *AI magazine*, vol. 13, no. 3, p. 57, 1992.

- [12] M. Wernick, Y. Yang, J. Brankov, G. Yourganov, and S. Strother, “Machine learning in medical imaging,” *Signal Processing Magazine, IEEE*, vol. 27, no. 4, pp. 25–38, 2010.
- [13] V. Rebolledo, “Plataforma para la extracción y almacenamiento del conocimiento extraído de los web data,” Master’s thesis, Universidad de Chile, 2008.
- [14] B. Abma, *Evaluation of requirements management tools with support for traceability-based change impact analysis*. PhD thesis, Master’s thesis, University of Twente, 2009.
- [15] B. Goethals, “Survey on frequent pattern mining,” *Univ. of Helsinki*, 2003.
- [16] R. Agrawal and R. Srikant, “Mining sequential patterns,” in *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pp. 3–14, IEEE, 1995.
- [17] M. Zaki, “Scalable algorithms for association mining,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 12, no. 3, pp. 372–390, 2000.
- [18] E. Liddy, “Natural language processing,” *Encyclopedia of Library and Information Science, 2nd Ed.*, 2001.
- [19] T. Kiss and J. Strunk, “Unsupervised multilingual sentence boundary detection,” *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.
- [20] C. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge University Press Cambridge, 2008.
- [21] M. Porter, “Snowball: A language for stemming algorithms,” 2001. Available on <http://snowball.tartarus.org/texts/introduction.html>.
- [22] S. DeRose, “Grammatical category disambiguation by statistical optimization,” *Computational Linguistics*, vol. 14, no. 1, pp. 31–39, 1988.
- [23] E. Brill, “A simple rule-based part of speech tagger,” in *Proceedings of the workshop on Speech and Natural Language*, pp. 112–116, Association for Computational Linguistics, 1992.
- [24] M. Marcus, M. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of english: The penn treebank,” *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.
- [25] T. U. of Pennsylvania, “Alphabetical list of part-of-speech tags used in the Penn Treebank Project.” [http://www.ling.upenn.edu/courses/Fall\\_2003/ling001/penn\\_treebank\\_pos.html](http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html), 2013. Seen on 10/02/2013.
- [26] S. Abney, “Parsing by chunks,” *Principle-based parsing*, vol. 44, pp. 257–278, 1991.
- [27] J. Carroll, T. Briscoe, and A. Sanfilippo, “Parser evaluation: a survey and a new proposal,” in *Proceedings of the 1st International Conference on Language Resources and Evaluation*, pp. 447–454, 1998.
- [28] J. Perkins, *Python Text Processing with NLTK 2.0 Cookbook*. Packt Publishing, 2010.

- [29] E. 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-Volume 7*, pp. 127–132, Association for Computational Linguistics, 2000.
- [30] D. Park and S. Kim, “The effects of consumer knowledge on message processing of electronic word-of-mouth via online consumer reviews,” *Electronic Commerce Research and Applications*, vol. 7, no. 4, pp. 399–410, 2009.
- [31] H. Shin, D. Hanssens, K. Kim, and B. Gajula, “Impact of positive vs. negative e-sentiment on daily market value of high-tech products,” 2011. Working paper.
- [32] F. Zhu and X. Zhang, “Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics,” *Journal of Marketing*, vol. 74, no. 2, pp. 133–148, 2010.
- [33] N. Kobayashi, K. Inui, and Y. Matsumoto, “Opinion mining from web documents: Extraction and structurization,” *Information and Media Technologies*, vol. 2, no. 1, pp. 326–337, 2007.
- [34] comScore/the Kelsey group, “Online consumer-generated reviews have significant impact on offline purchase behavior.” Press Release, November 2007. <http://www.comscore.com/press/release.asp?press=1928>.
- [35] Avenue A | Razorfish Digital, “Digital consumer behavior study,” 2007. Available on <http://cdn.blogosfere.it/iab/images/DigConsStudy.pdf>.
- [36] K. Dave, S. Lawrence, and D. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, ACM, 2003.
- [37] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Verlag, 2007.
- [38] L. Ku, Y. Liang, and H. Chen, “Opinion extraction, summarization and tracking in news and blog corpora,” in *Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*, no. 2001, 2006.
- [39] M. Hu and B. Liu, “Opinion extraction and summarization on the web,” in *Proceedings Of The National Conference On Artificial Intelligence*, vol. 21, p. 1621, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [40] A. Harb, M. Plantié, G. Dray, M. Roche, F. Trouset, and P. Poncelet, “Web opinion mining: How to extract opinions from blogs?,” in *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pp. 211–217, ACM, 2008.
- [41] A. Bifet and E. Frank, “Sentiment knowledge discovery in twitter streaming data,” in *Discovery Science*, pp. 1–15, Springer, 2010.



- [42] J. Read, “Using emoticons to reduce dependency in machine learning techniques for sentiment classification,” in *Proceedings of the ACL Student Research Workshop*, pp. 43–48, Association for Computational Linguistics, 2005.
- [43] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining,” in *Proceedings of LREC*, vol. 2010, 2010.
- [44] D. Jurafsky, J. Martin, A. Kehler, K. Vander Linden, and N. Ward, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, vol. 2. Prentice Hall New Jersey, 2000.
- [45] F. Balbachean and D. Dell’Era, “Análisis automatizado de sentimiento en textos breves de la plataforma twitter,” 2012.
- [46] H. Kim, K. Ganesan, P. Sondhi, and C. Zhai, “Comprehensive review of opinion summarization,” 2011. Unpublished, available in <https://www.ideals.illinois.edu/handle/2142/18702>.
- [47] Y. Lu, C. Zhai, and N. Sundaresan, “Rated aspect summarization of short comments,” in *Proceedings of the 18th international conference on World wide web*, pp. 131–140, ACM, 2009.
- [48] A. Popescu and O. Etzioni, “Extracting product features and opinions from reviews,” in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 339–346, Association for Computational Linguistics, 2005.
- [49] M. Hu and B. Liu, “Mining opinion features in customer reviews,” in *Proceedings of the National Conference on Artificial Intelligence*, pp. 755–760, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.
- [50] N. Archak, A. Ghose, and P. Ipeirotis, “Show me the money!: deriving the pricing power of product features by mining consumer reviews,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 56–65, ACM, 2007.
- [51] R. Decker and M. Trusov, “Estimating aggregate consumer preferences from online product reviews,” *International Journal of Research in Marketing*, vol. 27, no. 4, pp. 293–307, 2010.
- [52] A. Cadilhac, F. Benamara, and N. Aussenac-Gilles, “Ontolexical resources for feature based opinion mining: a case-study,” in *23rd International Conference on Computational Linguistics*, p. 77, 2010.
- [53] L. Zhao and C. Li, “Ontology based opinion mining for movie reviews,” *Knowledge Science, Engineering and Management*, pp. 204–214, 2009.
- [54] E. Vallés Balaguer, P. Rosso, A. Locoro, and V. Mascardi, “Análisis de opiniones con ontologías,” *Polibits*, no. 41, pp. 29–36, 2010.

- [55] V. Hatzivassiloglou and J. Wiebe, “Effects of adjective orientation and gradability on sentence subjectivity,” in *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pp. 299–305, Association for Computational Linguistics, 2000.
- [56] E. Riloff and J. Wiebe, “Learning extraction patterns for subjective expressions,” in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 105–112, Association for Computational Linguistics, 2003.
- [57] H. Yu and V. Hatzivassiloglou, “Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences,” in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, Association for Computational Linguistics, 2003.
- [58] E. Riloff, S. Patwardhan, J. Wiebe, *et al.*, “Feature subsumption for opinion analysis,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 440–448, Association for Computational Linguistics, 2006.
- [59] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin, “Learning subjective language,” *Computational linguistics*, vol. 30, no. 3, pp. 277–308, 2004.
- [60] T. Wilson, J. Wiebe, and R. Hwa, “Just how mad are you? finding strong and weak opinion clauses,” in *Proceedings of the National Conference on Artificial Intelligence*, pp. 761–769, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.
- [61] T. Wilson, J. Wiebe, and R. Hwa, “Recognizing strong and weak opinion clauses,” *Computational Intelligence*, vol. 22, no. 2, pp. 73–99, 2006.
- [62] 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.
- [63] P. 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*, pp. 417–424, Association for Computational Linguistics, 2002.
- [64] X. Ding, B. Liu, and P. Yu, “A holistic lexicon-based approach to opinion mining,” in *Proceedings of the international conference on Web search and web data mining*, pp. 231–240, ACM, 2008.
- [65] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, ACM, 2004.
- [66] 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, 2007.
- [67] K. Shein and T. Nyunt, “Sentiment classification based on ontology and svm classifier,” in *Communication Software and Networks, 2010. ICCSN’10. Second International Conference on*, pp. 169–172, IEEE, 2010.

- [68] A. Ghose and P. Ipeirotis, “Designing novel review ranking systems: predicting the usefulness and impact of reviews,” in *Proceedings of the ninth international conference on Electronic commerce*, pp. 303–310, ACM, 2007.
- [69] C. Cardie, J. Wiebe, T. Wilson, and D. Litman, “Combining low-level and summary representations of opinions for multi-perspective question answering,” in *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*, pp. 20–27, 2003.
- [70] G. Carenini, R. Ng, and A. Pauls, “Multi-document summarization of evaluative text,” in *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 305–312, 2006.
- [71] H. Nishikawa, T. Hasegawa, Y. Matsuo, and G. Kikui, “Optimizing informativeness and readability for sentiment summarization,” in *Proceedings of the ACL 2010 Conference Short Papers*, pp. 325–330, Association for Computational Linguistics, 2010.
- [72] Y. Seki, K. Eguchi, N. Kando, and M. Aono, “Opinion-focused summarization and its analysis at duc 2006,” in *Proceedings of the Document Understanding Conference (DUC)*, pp. 122–130, 2006.
- [73] V. Stoyanov and C. Cardie, “Partially supervised coreference resolution for opinion summarization through structured rule learning,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 336–344, Association for Computational Linguistics, 2006.
- [74] M. Hu and B. Liu, “Opinion extraction and summarization on the web,” in *Proceedings of the national conference on artificial intelligence*, vol. 21, p. 1621, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [75] B. Liu, M. Hu, and J. Cheng, “Opinion observer: analyzing and comparing opinions on the web,” in *Proceedings of the 14th international conference on World Wide Web*, pp. 342–351, ACM, 2005.
- [76] I. Titov and R. McDonald, “A joint model of text and aspect ratings for sentiment summarization,” *Urbana*, vol. 51, p. 61801, 2008.
- [77] K. Tateishi, T. Fukushima, N. Kobayashi, T. Takahashi, A. Fujita, K. Inui, and Y. Matsumoto, “Web opinion extraction and summarization based on viewpoints of products,” *Information Processing Society of Japan SIGNL Note*, vol. 93, pp. 1–8, 2004.
- [78] S. Lichtenstein and P. Slovic, *The construction of preference*. Cambridge University Press, 2006.
- [79] K. Scherer, “What are emotions? and how can they be measured?,” *Social science information*, vol. 44, no. 4, pp. 695–729, 2005.
- [80] S. O. Hansson and T. Grüne-Yanoff, “Preferences,” in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), winter 2012 ed., 2012.
- [81] W. Barnett and A. Serletis, “Consumer preferences and demand systems,” *Journal of Econometrics*, vol. 147, no. 2, pp. 210–224, 2008.

- [82] K. Lancaster, “A new approach to consumer theory,” *The journal of political economy*, vol. 74, no. 2, pp. 132–157, 1966.
- [83] R. Bartels, *The history of marketing thought*. Publishing Horizons Columbus, 1988.
- [84] N. Cartwright, *How the laws of physics lie*. Cambridge Univ Press, 1983.
- [85] I. Hacking, *Representing and intervening: Introductory topics in the philosophy of natural science*, vol. 355. Cambridge Univ Press, 1983.
- [86] C. Hockett, *A course in modern linguistics*. Macmillan, 1960.
- [87] T. Deacon, *The symbolic species: The co-evolution of language and the brain*. No. 202, WW Norton & Company, 1997.
- [88] R. Trask, *Language: the basics*. Psychology Press, 1999.
- [89] G. Box and N. Draper, *Empirical model-building and response surfaces*. John Wiley & Sons, 1987.
- [90] M. Bakhtin, *Speech genres and other late essays*, vol. 8. University of Texas Press, 1986.
- [91] I. Pollach, “Electronic word of mouth: A genre analysis of product reviews on consumer opinion web sites,” in *System Sciences, 2006. HICSS’06. Proceedings of the 39th Annual Hawaii International Conference on*, vol. 3, pp. 51c–51c, IEEE, 2006.
- [92] A. Vicentini, “The economy principle in language,” *Notes and Observations from Early Modern English Grammars. Mots, Palabras, Words*, vol. 3, pp. 37–57, 2003.
- [93] V. Fromkin, R. Rodman, and N. Hyams, *An introduction to language*. Wadsworth Publishing Company, 2010.
- [94] R. Huddleston and G. Pullum, *A student’s introduction to English grammar*. Cambridge University Press, 2005.
- [95] J. Chevalier and D. Mayzlin, “The effect of word of mouth on sales: Online book reviews,” tech. rep., National Bureau of Economic Research, 2003.
- [96] N. Hu, L. Liu, and J. Zhang, “Do online reviews affect product sales? the role of reviewer characteristics and temporal effects,” *Information Technology and Management*, vol. 9, no. 3, pp. 201–214, 2008.
- [97] W. Moe and D. Schweidel, “Online product opinions: Incidence, evaluation, and evolution,” *Marketing Science*, vol. 31, no. 3, pp. 372–386, 2012.
- [98] R. Gauthier and S. Ponto, *Designing systems programs*. Prentice-Hall, 1970.
- [99] D. Parnas, “On the criteria to be used in decomposing systems into modules,” *Communications of the ACM*, vol. 15, no. 12, pp. 1053–1058, 1972.
- [100] A. A. Juanet, “Diseño y construcción de un prototipo funcional de sitio web adaptativo que permita comunicar la oferta turística,” Master’s thesis, Universidad de Chile, 2012.

- [101] Python Software Foundation, “The Python Programming Language Website.” <http://http://www.python.org/>, 2013. Seen on 10/02/2013.
- [102] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. O’Reilly Media, Incorporated, 2009.
- [103] Django Software Foundation, “The Django Framework.” <http://www.djangoproject.com/>, 2013. Seen on 10/02/2013.