Sentiment Mining in a Location-Based Social Networking Space

Semantically Oriented Rule-Based Reviews' Classification

by

DOMENICO CARLONE

A Thesis submitted to the University of Aalborg in partial fulfillment of the requirements for the degree of

Master of Science in Information Technology, Software Construction

Department of Software and Media Technology
Aalborg University Esbjerg

June 2011

Title: Sentiment Mining in a Location-Based Social Networking Space
Subtitle: Semantically Oriented Rule-Based Reviews' Classification

Author:
Domenico Carlone (email: carlone.domenico@gmail.com)
Master Student at Aalborg University Esbjerg, September 2009 to June 2011

Academic Supervisor:
Dr. Daniel Ortiz-Arroyo (email: do@es.aau.dk)
Department of Electronic Systems
Computational Intelligence and Security Laboratory
Esbjerg Institute of Technology, Aalborg University

Abstract

In this work we describe a system to perform sentiment classification based on an unsupervised linguistic approach that uses natural language processing techniques to extract individual words from reviews in social network sites. Our pattern-based method applies classification rules for positive or negative sentiments depending on its overall score calculated with the aid of SentiWordNet. Searching for the best classification procedure, we investigated several classifier models created from a combinations of different methods applied at word and review level; the most relevant among them has been then enhanced with additional linguistically-driven functionalities, such as spelling correction, emoticons, exclamations and negation detection. Furthermore, an empirical study on Word Sense Disambiguation has been conducted on a set of test sentences extracted from the SemCor Corpus. We defined two gloss-centered word sense disambiguation techniques which rely on overlaps and semantic relatedness calculated on disambiguated glosses' definitions provided by eXtended WordNet. Experimental results confirmed that Word Sense Disambiguation can improve sentiment classification performance; moreover, they indicated that all the words potentially carry emotions, including nouns.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sentiment Analysis or *Opinion Mining* is an emerging discipline regarding *Information Retrieval (Text Mining)* and *Natural Language Processing.*

Sentiment analysis consists in detecting the subjectivity and sentiments contained in general opinions; opinions are different from factual information; they don't have a factual nature but a subjective nature; facts are objective descriptions of an entity or an event and their attributes; opinions are expressions that describe emotion and feeling of individual people respect to the entity or the event [Liu10b].

Nowadays, Information Retrieval systems such as search engines can't search for sentiment; they can just retrieve factual information through some keywords [Liu].

Sentiment analysis represents a new knowledge resource after the advent of the World Wide Web.

Companies which want to perform marketing analysis in order to understand people's attitudes are moving toward this new direction. It permits them to reduce costs; since there is no need to employ consultants; and the automatic detection of opinions substitute surveys and questionnaires [NY03].

Also individuals are affected by this kind of revolution. Internet accomplishes the task of discovering trends. People desire to have suggestions, and it is known that a good measure of quality and truth is a profuse quantity of positive opinions. Whatever they are searching for, whether it is products, services, events, political topics, or movies, they are always hunting for positive reviews. For example, before travelling if there is the necessity to book a hotel it is common to verify its rate through feedbacks. As it is reported in [Liu10b], earlier *"the Web, when an individual needed to make a decision, he/she typically asked for opinions from friends and families". "Now if one wants to buy a product, he/she is no longer limited to asking his/her friends and families because there are many product reviews on the Web which give opinions of existing users of the product".*

Generally people leave their ideas on forums, group discussions, and blogs. However, in this Web 2.0 era, the most widely exploited way to share contents is the social network.

We present in this work an algortihm for mining opinions considering some social networks.

Of course social networks such as Facebook provide several contents and not just general opinions; but there are also social networks, like Foursquare, Yelp, Qype, Where, CitySearch, that take place in a specific domain and present a set of related feedbacks. The previously mentioned social networks mainly concerned with describing "interesting" places within cities. In these social networks sites users post opinions about clubs, events or restaurants and some of their features such as food quality, customer satisfaction or atmosphere. This thesis presents a system capable of collecting and classifying user's opinions by identifying their type, or better their semantic orientation.

As explanation, switching back to sentiment analysis, terms can be classified by their polarity in positive and negative.

For the purpose of this procedure called sentimental classification some classical machine learning techniques can be employed.

Alternatively, there is another way to conduct this procedure by constructing a rule-based classifier, also called lexicon-based, that applies a Natural Language Processing (NLP) approach and some linguistically-driven principles; it works on a lexicon of subjective terms and on a set of semantic rules; in this manner the accuracy of the classification may be improved.

Opinions can also have a strength of attitude, for either orientation or subjectivity [Esu06]. The algortihm should be able to associate a degree of positivity or negativity with each comment in order to obtain a ranked list of the best reviewed places.

Tools and lexical resources such as SentiWordNet, are exploited in this work.

The effectiveness of the proposed system is evaluated in terms of Precision, Recall, F-measure and overall Accuracy.

In future work the rule-based classifier may be compared with others such as Support Vector Machines, Naive Bayes or Maximum Entropy.

Chapter 2

Background

2.1 Problem Definition

"In a world in which millions of people write their opinions about any issue in blogs, news sites, review sites or social media, the distillation of knowledge from this huge amount of unstructured information is a challenging task. Sentiment Analysis and Opinion Mining are two areas related to Natural Language Processing and Text Mining that deal with the identification of opinions and attitudes in natural language texts."[1]

Natural Language Processing (NLP) is a *Computational Linguistics area* pertaining to computer manipulation of natural language. It concerns the extraction of syntactic and semantic information from natural language expressions. Sentiment mining involves NLP in the correct and automatic interpretation of natural language.

Text Mining belongs to *Data Mining's field.* Both consist in deriving patterns from data, where Text Mining concerns finding hidden text patterns on natural language's texts. Related tasks include text classification or clustering, text summarization, other than sentiment analysis.

[1] 1st Workshop on Opinion Mining and Sentiment Analysis, at CAEPIA-TTIA, November 13, 2009, Seville, Spain, `http://sites.google.com/site/womsa09/`

"Opinion mining (Sentiment Mining, Opinion/Sentiment Extraction) attempts to make automatic systems to determine human opinion from text written in natural language. The main advantage is the speed; on average, humans process six articles per hour against the machine's throughput of 10 per second."[2]

Motivation and Sentiment Classification's Applications

The Web represents a huge container of opinionated content due to the ease of publishing on-line; opinions and reviews are easily posted, by people that have a minimum of technical knowledge, in review portals, newsgroup posts, blogs, internet forums or, more recently, in social networks. These data are commonly referred to as user-generated content and they usually come in an unstructured "free textual" form. For this reason we deal with *Text Mining*. Nowadays, unstructured text represents the majority of information available to a particular research.

An application for Text Mining is to contribute in the automatic classification of texts; text classification is commonly based on extracted information about its content. In this work, reviews retrieved from social networks are classified on the base of the presence of certain terms that are likely to express a sentiment; this process is called *Sentiment or Opinion-Oriented Classification*; given an opinionated piece of text, a review in our specific case, the goal is to classify the opinion as belonging to one of two opposing sentiment polarities: positive or negative [Mej10]. In order to apply the classification method, the data is prepared using *Natural Language Processing*.

Sentiment Classification falls under *Sentiment Analysis* which consists in tracking sentiments expressed on some target entities. In this report we consider interesting places. In general, an entity can be a product, person, event, organization, or topic [Liu10b]. "Tracking sentiments" means to understand what people likes and dislikes. In this Web 2.0 era, the online "word-of-mouth" provides a huge amount of this kind of information; users are affected

[2]Dr. Alaa El-Halees - Opinion Mining Seminar, September 9, 2008, Department of Computer Science, Islamic University of Gaza.

by opinions of others; for example, they use blogs or other portals to monitor trends; and vendors, companies or product manufacturers are interested in people thinking; for instance, they try to discover consumers satisfaction about products on the Internet.

Activity in *Sentiment Analysis* is growing on large scale area including politics (e.g., understanding what voters are thinking (political opinions) in estimating political polling), business (e.g., marketing research), blog and social media analysis (e.g., analyzing blog sentiments about movies in order to correlate them with sales). At the same time, *Opinion-Oriented Classification* find employ, as a sub-component technology, for example, in Search Engines, solving the issue of searching for subjective web pages, or better web pages regarding opinionated content; or in Recommender Systems, where positive reviews can be considered a recommendation. Moreover, sentiment-aware applications include other emotion-aware Information Retrieval systems, such as *Opinion Question Answering Systems* which are able to successfully answer questions about people's opinions (e.g., What is the international reaction to the reelection of Robert Mugabe as President of Zimbabwe? African observers generally approved (positive) of his victory while Western governments denounced (negative) it) [WWH09].

This report discusses the existing works on opinion mining and sentiment classification of reviews, and describes and evaluates a rule-based technique used for a reviews' classification process in a Location-Based Social Networking domain.

2.2 Sentiment Analysis or Opinion Mining

Sentiment Analysis or Opinion Mining consists in trying to detect subjectivity and sentiment contained in general opinions, expressed in natural language; opinions are different from factual information; they don't have a factual nature but a subjective nature; facts are objective descriptions of an entity or an event and their attributes; opinions are expressions that describe emotion and feeling of individual people respect to the entity or the event

[Liu10b].

Analysis of opinions in text can be seen as a two steps process. It firstly consists in *identifying* the *opinion expressions*; where, as we will see in a further example, a text can contain more than one of them. It secondly involves the identification of [Esu06]:

- sentiment properties of opinions, such as orientation or attitude and strength;
- who is expressing them, also known as opinion holder;
- their target.

Esuli and Sebastiani [GJM10] have organized the problem of identifying sentiment in text into three subtasks:

1. *Determining subjectivity*, as in deciding whether a given text has a factual nature (objective) or expresses an opinion (subjective).
2. *Determining orientation* (or *polarity*), with the goal of discover if a given subjective text expresses a positive or negative opinion.
3. *Determining the strength of orientation*, where it can be expressed, for example, by an adjective (e.g., weakly positive or strongly negative) or by a numerical value (e.g., a positive or negative score ranging in the interval [0,1]).

An important task that is complementary to sentiment identification is the discovery of the *target* on which an opinion has been expressed. Targets are objects or entities, such as products, services, individuals, organizations, events, topics; opinions can refer to their features: components (or parts) and attributes (or properties). *"Such information is not discovered by sentiment and subjectivity classification;"* however it is important to understand (more in the specific) what is liked or disliked about an entity [Liu10a].

Let's report an example to clarify all the previous concepts, introducing a review used in [Liu10a]:

"(1) *I bought an iPhone 2 days ago.* (2) *It was such a nice phone.* (3) *The touch screen was really cool.* (4) *The voice quality was clear too.* (5) *However, my mother was mad with me as I did not tell her before I bought it.* (6) *She also thought the phone was too expensive, and wanted me to return it to the shop. ...* "

It can be noticed that several opinions, together with more than one sentiment orientation, are expressed in this review; sentences (2), (3) and (4) represent three positive opinions, while sentences (5) and (6) represent negative opinions.

The target on which opinions are referring to is also changing through the sentences. For example, the opinion in sentence (2) regards the iPhone as a whole; opinions in sentences (3) and (6) regard the "touch screen" component and "price" property of the iPhone respectively.

Moreover, the persons who are expressing opinions (holders) are different in the review. In sentences (1) and (2) the holder is the author of the review ("I"), but in sentences (5) and (6) it is "my mother".

Chapter 3

Related Works

In the field of Sentiment Analysis, many studies have been carried out on Sentiment-based Classification; anyway, none or few of them regard the social network review domain. The area of interest most closely related to this research is movie review classification.

In this chapter we first discuss on Sentiment Analysis and Natural Language Processing aspects of previous research; then, we concentrate on review classification and the use of SentiWordNet as lexical resource.

Looking at related works and methods adopted, a distinction can be made between machine learning and linguistic (or semantic orientation) approaches.

In our study, Sentiment Analysis is performed at sentence level (clause-level or phrase-level Sentiment Analysis) and Sentiment Classification is based on a linguistic approach; our pattern-based method applies a classification rule according to which each review is classified as positive or negative depending on its overall sentiment score, calculated with the aid of SentiWordNet.

Word Sense Disambiguation research is also reported.

3.1 Overview

Sentiment Analysis deals with the computational study of sentiment in natural language text.

Majority of works in the area focus on assigning sentiments to documents; some other researches concern about more specific tasks, such as finding the sentiments of words [HM97] or searching for subjective expressions [WWH05].

Sentiments, and their relatives texts, can be distinguished, through Sentiment Classification, in opinionated (subjective) or factual (objective) [WWH05]; at the same time, subjective texts can be divided in containing positive or negative sentiments.

Two approaches have mainly been applied to sentiment classification: machine learning [PLV02] and semantic orientation [Tur02]. The last one is also identified as linguistic; it is a rule-based (or pattern-based) approach that implies Natural Language Processing [NY03] and, sometimes, the use of external lexical resources; the first one employs machine learning algorithms, such as Naive Bayes (NB), Maximum Entropy (ME) or Support Vector Machine (SVM).

Sometimes the two approaches are combined in a hybrid solution, like in [DZC10], [Den08] and [PT09].

In order to apply a machine learning approach, the classifier must be trained on a set of known data; in contrast, the semantic orientation approach does not require prior training; while in the first case the classifier learns its method on already classified data, in the second case the polarity orientation of a document or review is inferred using some linguistic heuristics from the polarity of its words.

For the previous reason, machine learning and linguistic are also referred to as "supervised learning" and "unsupervised learning" approaches respectively.

The difference between them stands in the training phase that, if executed on a representative corpus for the domain, helps to achieve better results; therefore, the learning algorithm adapts to the different characteristics of the domain under consideration while rules are fix for all domains [Den09];

consequently, the machine learning approach tends to be more accurate, while the semantic orientation approach has better generality [DZC10].

"The benefit of the rule-based approach is that no training material is required" [Den09]. The gathering of such a corpus is usually arduous; training material is *"normally sparse for different languages or is even unavailable"* [Den08]; thereby, human manual classification of huge amount of data may be required to compose the training set. On the other hand, semantic orientation approaches necessitate language specific lexical and linguistic resources, hard and time consuming to produce [Den08]. In terms of timing, the machine learning approach requires a significant amount of time to train the model while the semantic orientation approach can be used in real-time applications [CZ05].

As we have previously mentioned, the linguistic approach implies NLP techniques; usually, phrases containing opinions are extracted looking at predefined part-of-speech patterns; in [Tur02], for example, Turney uses part-of-speech tagging to extract two-words phrases containing at least one adjective or one adverb from the review, in order to estimate the semantic orientation of the review, averaging the semantic orientation scores of the phrases within. Turney's work and others such as [HM97] assert high correlation between the presence of adjectives and sentence subjectivity. Other studies like [PLV02] demontrate that also other parts of speech such as nouns and verbs can be significative flags of sentiment [PL08].

In the same study [PLV02], Pang et al. examine three different machine learning approaches for sentiment classification: Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy. Since these approaches work well in text categorization, the aim of the research was to consider and investigate the problem of sentiment classification of movie reviews as "topic-based" categorization between positive and negative. They evaluated the previously listed supervised learning algorithms using bag-of-words features common in text mining research, obtaining best performance using Support Vector Machines in combination with unigrams, reaching a maximum accuracy of 83%. Produced results confirmed that machine learning methods are usually better in comparison to human generated baselines in sentiment classification.

In text classification, and sentiment classification as well, different kind of feature selection patterns can be taken into account, althought unigrams seems to be the most effective for machine learning approaches; for instance, other n-grams features such as bigrams (couple of words) or trigrams; where if more words are considered then more context is gained. Reviews can be seen as feature vectors where different feature weighting methods can be applied, including Feature (or term) Presence (FP), Term Frequency (TF) and TF-IDF. Pang et al. study [PLV02] and [OK09] found presence (FP), rather than frequency, to be the most accurate feature weighting method; an explanation is given by Pang et al. in [PL08] where they compare "topic-based" with sentiment classification: *"While a topic is more likely to be emphasized by frequent occurrences of certain keywords, overall sentiment may not usually be highlighted through repeated use of the same terms"*.

Two types of techniques have been used in sentiment classification based on semantic orientation approach: corpus-based and dictionary-based techniques.

"The corpus-based techniques aim to find co-occurrence patterns of words to determine their sentiments. Different strategies are developed to determine sentiments" [DZC10]. For example, Turney [Tur02] calculated a phrase's semantic orientation considering Point-wise Mutual Information (PMI, based on probability of collocations (Church and Hanks, 1989)) between terms within it and two reference words "excellent" and "poor" representative of the positive and negative polarity.

Dictionary-based techniques utilize dictionaries and sentiment lexicons, giving information about semantic relations between words and terms' sentiment properties, to determine overall sentiment of opinions. WordNet is a semantic database resource that helps to discover relations between english words; SentiWordNet is a sentiment lexicon built upon WordNet that has been used in recent sentiment classification studies.

Some important researches employing SentiWordNet for sentiment classification are described in the following Section 3.2, while in Figure 3.2 a summary of existing research works in sentiment analysis are reported.

Author	Objectives	N-Gram	Model	Data Source	Eval. Method	Data Set	T_r	T_e	Accuracy	Precision	Recall	F_1
Gamon (2004)	Assign docs sentiments using 4-point scale		SVM	customer feedback	10 fold cross validation (1 vs 4)	N/A	36796	4084	77.5	N/A	N/A	N/A
					10 fold cross validation (1,2 vs 3,4)	N/A	36796	4084	69.5	N/A	N/A	N/A
Pang & Lee (2005)	Assign docs sentiments using 3-point or 4-point scale		SVM, Regression, Metric Labeling	movie reviews	10 fold cross validation (3 point-scale)	5006	N/A	N/A	66.3	N/A	N/A	N/A
					10 fold cross validation (4 point scale)	5006	N/A	N/A	54.6	N/A	N/A	N/A
Choi et al. (2005)	Extract the sources of opinions, emotions and sentiments		CRF and AutoSlog	MPQA corpus	10 fold cross validation	N/A	135	400	N/A	70.2–82.4	41.9–60.6	59.2-69.4
Wilson et al. (2005)	Assign expressions +/-/both/neutral		BoosTexter	MPQA corpus	10 fold cross validation: polar/neutral	13183 expressions	N/A	N/A	73.6-75.9	68.6-72.2 / 74.0-77.7	45.3-56.8 / 85.7-89.9	55.7-63.4 / 80.7-82.1
					10 fold cross validation: +/-/both/neutral	13183 expressions	N/A	N/A	61.7-65.7	55.3-63.4 / 64.7-72.9 / 28.4-35.2 / 50.1-52.4	59.3-69.4 / 80.4-83.9 / 9.2-11.2 / 30.2-41.4	61.2-65.1 / 73.1-77.2 / 14.6-16.1 / 37.7-46.2
König & Brill (2006)	Assign docs sentiments		Pattern-based, SVM, Hybrid	movie reviews	5 fold cross validation	1000(+) 1000(-)	N/A	N/A	>91	N/A	N/A	N/A
				customer feedback	5 fold cross validation	N/A	30000	10000	<72	N/A	N/A	N/A

Figure 3.1: Existing work in sentiment analysis [PT09].

Author	Objectives	N-Gram	Model	Data Source	Eval. Method	Data Set	T_r	T_e	Accuracy	Precision	Recall	F_1
Hatzivassiloglou & McKeown (1997)	Assign adjectives +/-	N/A	non-hierarchical clustering	WSJ corpus	N/A	657adj(+) 679adj(-)	N/A	N/A	78.1–92.4	N/A	N/A	N/A
Pang et al. (2002)	Assign docs sentiments	uni- & bi-grams	NB, ME, SVM	movie reviews	3-fold cross validation	700(+) 700(-)	N/A	N/A	77–82.9	N/A	N/A	N/A
Turney (2002)	Assign docs sentiments	N/A	PMI-IR	automobile, bank, movie, travel reviews	N/A	240(+) 170(-)	N/A	N/A	65.8-84	N/A	N/A	N/A
Yi et al. (2003)	Assign topics sentiments	-	NLP, Pattern-based	digital camera, music reviews	N/A	735(+) 4227(-)	N/A	N/A	85.6	87	56	N/A
				petroleum, pharmaceutical Web pages	N/A	N/A	N/A	N/A	90-93	86-91	N/A	N/A
Nasukawa & Yi (2003)	Assign topics sentiments	-	NLP, Pattern-based	Web pages	N/A	118(+) 58(-)	N/A	N/A	94.3	N/A	28.6	N/A
				camera reviews	N/A	255	N/A	N/A	94.5	N/A	24	N/A
Dave et al. (2003)	Assign docs sentiments	uni-, bi- & trigrams	Scoring, Smoothing, NB, ME, SVM	product reviews	macro-averaged	N/A	13832(+) 4389(-)	25910(+) 5664(-)	88.9	N/A	N/A	N/A
							2016(+) 2016(-)	224(+) 224(-)	85.8	N/A	N/A	N/A
Hiroshi et al. (2004)	Assign topics sentiments	-	NLP, Pattern based	camera reviews	N/A	200	N/A	N/A	89-100	N/A	43	N/A
Pang & Lee (2004)	Assign docs sentiments	unigrams	NB, SVM	movie reviews	10-fold cross validation	1000(+) 1000(-)	N/A	N/A	86.4-87.2	N/A	N/A	N/A
Kim & Hovy (2004)	Assign expressions sentiments		Probabilistic based	DUC corpus	10-fold cross validation	N/A	231 adjectives	N/A	75.6-77.9	N/A	97.8	N/A
							251 verbs	N/A	79.1-81.2	N/A	93.2	N/A
							N/A	100 sentences	81	N/A	N/A	N/A

Figure 3.2: Existing work in sentiment analysis (continued) [PT09].

3.2 Deepening of Research Works

3.2.1 SentiWordNet and Sentiment Classification of Reviews

"Sentiment classification is an opinion mining activity concerned with determining what, if any, is the overall sentiment orientation of the opinions contained within a given document. It is assumed in general that the document being inspected contains subjective information" [OT09]. Opinions can be classified by their orientation or score, as falling under two opposite polarities: positive and negative.

Several researches present the results of applying the SentiWordNet lexical resource to the problem of automatic sentiment classification; some of them are described as follows.

Pera, Qumsiyeh and Ng [MSPN10] introduced a domain independent sentiment classifier which categorizes reviews on the base of their semantic, syntactic, and sentiment content. The proposed classifier, in order to calculate the overall sentiment score of a review, first determines the polarity score of each word contained in it; thereafter, it calculates the review's sentiment orientation by subtracting the sum of its negative words' scores from the sum of its positive words' scores.

Thet, Na, Khoo, et al. [TNKS09] proposed a linguistic approach for sentiment analysis of message posts on discussion boards, in which they perform a clause-level sentiment analysis; they firstly calculate the prior words' sentiment scores, employing SentiWordNet in combination with a movie review domain specific lexicon built on purpose; then, they determine the contextual sentiment score for each clause analyzing grammatical dependencies of words, through dependency trees, and handling pattern-rules, such as Negation.

Denecke [Den08] introduced a methodology for determining polarity of documents within a multilingual context. Before to proceed with classification of text as belonging to a positive or negative sentiment class, a document is translated in English by making use of a translation software if it is written in a different language. Sentiment Classification involved three methods: SentiWordNet Classifier with classification Rule (Rule-Based), SentiWordNet Classifier with Machine Learning approach, LingPipe's text Classification algorithm. Best results were achieved using machine learning techniques.

In **[Den09] Denecke** executed a similar study to the previous, testing rule-based and machine learning approaches in a multi-domain, instead of multilingual, classification scenario. Results confirmed that the lexicon-based approach that make use of SentiWordNet achieved *"only results of very limited accuracy"* compared to the machine learning method. *"Nevertheless, the results show that SentiWordNet can be used for classifying documents of different domains according to their sentiment"*.

Some few studies have combined semantic orientation and machine learning approaches to improve Sentiment Classification performance. **Yan Dang, Zhang, and Chen [DZC10]** combined the two approaches into one framework. The lexicon-based method is "enhanced" using the words with semantic orientations as *"an additional dimension of features (referred to as "sentiment features") for the machine learning classifiers"*.

Ohana and Tierney [OT09] compared two approaches that assess the use of SentiWordNet to the task of document level sentiment classification of film reviews. In the first, the lexicon is applied *"by counting positive and negative terms found in a document and determining sentiment orientation based on which class received the highest score"*, similar to the methods presented in [PLV02] and [KI06]; thereby term scores are used to determine sentiment orientation. The second method uses SentiWordNet as a source of positive and negative features, in order to train a SVM supervised learning algorithm, reporting an improvement in accuracy.

3.2.2 Word Sense Disambiguation and Extended WordNet

"Word Sense Disambiguation (WSD) is an intermediate task of Natural Language Processing. It consists in selecting the appropriate meaning of a word given the context in which it occurs" [MBMP10].

Althought WSD constitutes an intermediate task in Polarity Classification, disambiguation errors can affect the classification quality.

Disambiguation research started in the **early 1960s** with manually created rules moving *"towards automatically generated rules based on disambiguation evidence derived from existing corpora available in machine readable form"* [San96]; since the **mid 1980s**, large-scale lexical resources such as dictionaries, thesauri, and corpora became widely available [Esu08a]. The first work that made use of a machine-readable dictionary was the Lesk Algorithm [Les86] (1986); which performed a disambiguation of two or more words by finding the pair of dictionary senses with the greatest word overlap in their dictionary definitions, since each definition was considered as a bag of words.

Nowadays, looking at the different corpora types employed, approaches to automatic disambiguation can be broadly classified in: *knowledge-based (or knowledge-driven) methods*, which rely primarily on dictionaries, thesauri or semantic lexicons, without using any corpus evidence; *corpus-based or supervised methods* (i.e. data-driven methods based on statistics), that make use of sense-annotated corpora, manually constructed by disambiguating words in a corpus, in order to train a disambiguator; *unsupervised methods*, according to which, disambiguators work directly on unannotated corpora without the use of any external resource or training, and words' senses are individualized employing clustering.

In **[ASPPBL06] Sánchez et al.** applied a *clustering* algorithm to the disambiguation process reaching 47% of Recall; starting from a distribution of all possible senses of the ambiguous words, clusters that best match the context were selected while the other were discarded, until the selected clusters disambiguated all words. In **[LSM95] Li et al.** used a set of *heuristic rules* and defined *patterns*; the disambiguation was guided by semantic similarity between words which could result in a strength at different levels.

Even if Word Sense Disambiguation has been addressed by many researchers, *" no satisfactory results are reported. Rule based systems alone can not handle this issue due to ambiguous nature of the natural language. Knowledge-based systems are therefore essential to find the intended sense of a word form"* [KB09].

"The wide availability of WordNet as a concept hierarchy has led to the development of a number of approaches to disambiguation based on exploiting its structure" [PBP03].

Researches like **[PBP]** rely on *WordNet* and relationships among synsets of the words' concepts in order to perform the WSD; others **[RygBP05]** make use of the *Web* as knowledge source for disambiguation, together with WordNet, searching for syntactic or text-proximity relations between words.

Magnini, Strapparava et al [MSPG02] developed a lexical resource called WordNet Domains, being an extension of WordNet, which binds each WordNet synset to a set of established of Domains. They investigated the role of *domain* information in Word Sense Disambiguation.
They demonstrated as their WSD algorithm can be based on domain information, in addition to senses; where the use of domain annotation for every synset, in the form of domain labels, represents a *"useful way to establish semantic relations among word senses"*.

In a similar study [KB08], **Kolte and Bhirud** proposed a scheme for determining the domain of a target word in a text, considering the *domain* of the surrounding words in the local context; the sense corresponding to the domain individualized with the aid of WordNet Domains is taken as the correct sense.

Some other methods apply an *enriched gloss centered WSD* inspired by Lesk's algorithm; in **[RPB04]**, for example, glosses' descriptions are personalized being generated using glossaries or encyclopedias, and glosses' comparisons are based on metrics as Jaccard or Cosine similarity.

More works can be considered as *adaptations of* the original *Lesk algorithm*, using WordNet instead of a standard dictionary as glosses' source, in order to take advantage of the network of relations provided.

In **[EG04]** the authors used *WordNet* in combinations with the *Lesk algorithm* in order to include in the *overlap* comparison not only the terms contained in the dictionary's definitions but also the terms contained in the definitions of the two nearest WordNet *hypernyms* of the word to disambiguate.

In **[BP02]** and **[BP03]**, a window of context words is defined and words are compared in pairs looking at the *overlap* between their glosses. The algorithm compare glosses associated with *hypernyms (i.e. parent), hyponyms (i.e. child), holonyms (i.e. is–a–part–of), meronyms (i.e. has–part) or attributes* of each word in the pair.

In **[PBP03]** and **[PBP]** the Lesk method is generalized *"by creating an algorithm that can perform disambiguation using any measure that returns a relatedness or similarity score for pairs of word senses"*; nine different *measures of semantic relatedness* can be plugged into the Lesk algorithm in place of gloss overlaps.

Extended WordNet is a *disambiguated sense inventory* built upon WordNet. It refers to the paper **[art04]** where it is described as the WordNet glosses are semantically disambiguated basically on a set of heuristics, reaching an overall precision of 86%.

[NB07] is always inspired by Lesk but the disambiguation involves looking for *overlaps* on *synsets' sense tagged glosses* relied to *Extended WordNet.* Given a target word (i.e. the word to disambiguate), all the terms present in the same sentence, in the preceding sentence and in the succeeding sentence contribute to the disambiguation; also the meanings of words that are connected to the target term through pre-selected WordNet relationships are taken into account. *"The system has been evaluated on the first 10 Semcor2.0 files and produces a precision of 85.9%, and 62.1% recall"*.

The **idea behind our approach** is inspired by several works [BP03] [NB07] [BP02] and it is described in Section 5.3.2.

We can conclude saying that the task of Word Sense Disambiguation has been demontrated as being relevant for Sentiment Classification. The advantages are given in the superiority of the results if the disambiguation is correct; at the same time, it is easy to fall into errors which can significantly affect the classification quality. This provides further motivation to study in depth this problem.

Chapter 4

Analysis

In this chapter, phases and steps that are implicated in an **opinion mining** study are analyzed.

We can say we can intend this work to be composed of two main stages: *opinion extraction* and *opinion classification*.

The reason of the previous assertion is that in order to process opinions it was initially necessary to collect them.

Usually sentiment analysis experiments can involve the use of available datasets[1], but looking at the literature they mainly pertain to movie reviews. The particular domain of social networks resulted unexplored and no data set was found; therefore, the data extraction phase was also a challenging and laboured part of this work.

A third phase called *opinion visualization* may be necessary in the future.

4.1 Data

Nowadays social networks agree to sharing of information, and opinions, among friends or, more generally, a community of users.

There are social networks, like "Foursquare", "Yelp", "Qype", "Where", "Brighkite" or "CitySearch", that take place in a specific domain and present

[1]Sentiment polarity and subjectivity data sets can be found, for example, at `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

information about interesting places; they allow to discover new places or learn about the places own friends frequent; in these social networks users can give opinions about clubs, events or restaurants, and their features, such as food or atmosphere.

Some of the previously listed social networks provide APIs to access their databases, but not every of them (e.g., no "Brighkite"); moreover, opinions are not all usable because of different from English languages by which they are written.

We decided to work with "Yelp" and "Foursquare" in this study; where the social networks can be seen as the data sources and our data consist in geo-coded place reviews collected from them.

Yelp's reviews are associated to a rating expressed on a 5-point scale, with 1 being the most negative and 5 being the most positive.

We decided to convert the favorability expressed for each review by its rating into text polarity (corresponding to one of the three sentiment categories: positive, negative, or neutral), in order to use it during training and testing of the Sentiment Classification.

Each review having a rating (i.e. number of stars (1 to 5)) of 1 or 2 and 4 or 5 is labeled as negative and positive respectively; opinions marked with three stars are considered to be objective or neutral.

As it is suggested in [PLV02] and [PL08], rating, for instance in terms of number of stars, can be used as indicator of overall sentiment of reviewers avoiding manual annotation of data for supervised learning or evaluation purposes.

4.2 Opinion (Data) Extraction

Information Retrieval (IR) is concerned with identifying documents, among a collection, which are relevant to a given topic.

Sentiment Analysis is a subfield of IR that may require the identification and extraction of pieces of text where opinions are expressed, in working with documents and sentiment orientation or subjectivity as topic.

A field of Natural Language Processing devoted to this type of task is Information extraction (IE); its main function is to process natural language text in order to select specific pieces of information contained in it. The IE's goal is common to summarization or question answering systems; in opinion mining the process is identified as "opinion extraction", or as "opinion-oriented information extraction" when it refers to more specific tasks, like the extraction of particular aspects or features of such an entity [PL08].

In this study we don't work with documents but with subjective reviews that concentrate opinions in their short text; at this stage of the work we don't need to locate specific pieces of information within reviews, because we analyse them as a whole; anyway a "data", more than "opinion", "extraction" phase is necessary to collect reviews from the social networks under consideration: Yelp and Foursquare.

The unstructured text information extracted is then entered into a structured database to be used for further processing.

4.3 Opinion Classification: Data (Text) Mining

In this Section a description of which decisions have been taken along the *Opinion Mining* process is given.

A requirements analysis' phase is necessary in order to carry out an understanding of the desired behaviour of the system.

4.3.1 Requirements Elicitation

The Data (Text) Mining aim of this study is to classify reviews by its content as "positive" or "negative" (*Sentiment Classification*). This main task concerns activities like the identification of sentences within a review and the discovering of polarity of words contained in it.

An initial requirements envisioning/modeling phase was performed to outline a first idea of the sub-goals that should be involved in solving the problem of the Sentiment Classification.

A stack of user stories is reported next in a point-style list, to the previous purpose; user stories are brief and high-level usage requirements artifacts, different from, and smaller than, other requirements specifications such as use cases or usage scenarios; they are usually used in Extreme Programming by developers and customers to discuss and negotiate technical and business decisions about a software devolpment process, concerning small releases and several iterations; user stories are firstly elicitate to then be revisited. In this case study, in which no particular interactions occur between the system and the user, the "data miner" can be identified as being the main actor involved in the following user stories; the prerequisite for their execution must be the granted access to the database where there should be stored the reviews retrieved from the social networks.

We can say that as "data miner" we want:

a) to parse text in order to understand "grammatical structure" of opinions;

b) to "chunk and tag words" in order to establish part-of-speech of terms occuring;

c) to understand the semantic orientation (or polarity) of terms;

d) to establish strength of term attitude (either orientation or subjectivity), in terms of degrees of positivity, negativity, objectivity;

e) to discover relations between terms (context), and to optionally treat multi-word expressions;

f) to disambiguate words' semantic using their context (Word Sense Disambiguation);

g) to identify polarity shifters, or better, words that may shift a negative polarity to a positive one, and viceversa;

h) to extract some features or attributes on which opinions are expressed.

4.3.2 Requirements Analysis and Research Methods

Requirements Analysis has the purpose of determining which subset of elicited requirements are appropriate to be adressed in the specific release of the system.

Existing researches, about phases involved in solving the problem of Sentiment Classification, are examined in order to discover related and relevant solutions.

Lexical Resources or Sentiment Lexicons

A common point in Opinion Mining studies is the need to identify which lexical items (single word or multiword expressions) contribute to express sentiment in text. If related to the English language, such task can be accomplished by using external lexical resources [Esu08a].

As it was explained in Section 2.2, the problem of identifying sentiment in text can be expressed in terms of determining subjectivity and semantic orientation (or polarity). Lexicons that address the first of previous subtasks are identified as *subjectivity lexicons*; they provide lists of subjective words (subjectivity clues), such as the one introduced in [WWH05][2]. Other lexicons contribute to code prior polarity of words, such as Harvard General Inquirer (GI) [SDSO66][3], Micro-WNOp [CCD+07][4] and SentiWordNet [ES06] [BES10]; the first twos point out prior polarities together with indicators (i.e. adjectives) of term attitudes (e.g., "strong negative" or "weak positive"); SentiWordNet furnishes degrees of words' polarities within the range [0,1] referring not only to positivity and negativity but also to objectivity; therefore, in SentiWordNet, polarity scores express also strength of term subjectivity. Regarding the coverage of the language, General Inquirer consists of 4206 entries (1915 and 2291 words of positive and negative outlook respectively), Micro-WNOp corpus is composed of 1105 WordNet

[2] Available at http://www.cs.pitt.edu/mpqa

[3] Downloadable at `http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm` Documentation: `http://www.wjh.harvard.edu/~inquirer/homecat.htm`

[4] Documentation and download: `http://www.unipv.it/wnop`

synsets, and SentiWordNet assigns sentiment scores to each WordNet entry (more 80K of unique words).

Prior vs Contextual Polarity

The semantic orientation (or polarity) of a word *"might be said to generally bear when taken out of context"* [PL08]. However, context can influence a term attitude; where for context of a word we mean other lexical items surrounding it. A word may appear in a phrase that expresses a different polarity in context. Polarity of words can be, then, prior or contextual. A sub-concept of contextual polarity is target-specific polarity; also polarity shifters take role in influencing contextual polarity.

Polarity Shifters

"Besides bearing a negative or positive polarity, words can be polarity shifters. Negation is the most common form" [KFP09]; it can be applied near adjectives, verbs, nouns, reversing their polarity; the "not" in "this is not a bad joke" shifts the negative polarity of "bad joke" to a positive polarity [KFP09]; the same happens using "do" and "does not" together with verbs (e.g., "I do not like") or "no" before subjects/objects (e.g., "no one I liked")

Also conjunctions determine variations in polarity of linguistic expressions; a conjunction rule was stated in [HM97]: adjectives in *and* conjunctions usually have similar orientation, though *but* is used with opposite orientation (e.g., "elegant but expensive", "tasty and light"). Since in English *but* means contrary, *but* can be identified as an evidence for phases in which the opinion orientation before *it* and after *it* are opposite to each other; such phrases are usually referred to as *but-clauses.*

"Clearly negation words are important because their appearances often change the opinion orientation. However, negation words must be handled with care because not all occurrences of such words mean negation. For example, "not" in "not only ... but also" does not change the orientation direction" [Liu10b].

Also nouns, multiword expressions and verbs can assume the role of polarity shifters; like for example "lack of" in "lack of understanding", or "abate" in "abate the damage"; therefore, not only adjectives matter.

Moreover, there are lexical items that instead of reverse the polarity they modify the valence of a term, weakening or strengthening it; they are called *intensifiers* and *diminishers*; most of them are adverbs and they can act like, for instance, "rather" in "rather efficient" or "deeply" in "deeply suspicious" [PZ06].

In conclusion, an analysis of polarity shifters can significantly reduce errors in Sentiment Classification.

Word Sense Disambiguation

"Word Sense Disambiguation (WSD) is an intermediate task of Natural Language Processing. It consists in selecting the appropriate meaning of a word given the context in which it occurs" [MBMP10].

Words in Natural Language are polysemouses and, in different contexts, they may not have the same polarity because of the multiple meanings they can assume; for example "a cheap meal" expresses a positive sentiment if "cheap" means "low price" but negative if it means "low quality".

Althought WSD constitutes an intermediate task in Polarity Classification, disambiguation errors can affect the classification quality.

Disambiguation research started in the early 1960s with manually created rules moving *"towards automatically generated rules based on disambiguation evidence derived from existing corpora available in machine readable form"* [San96]; since the mid 1980s, large-scale lexical resources such as dictionaries, thesauri, and corpora became widely available [Esu08a]. The first work that made use of a machine-readable dictionary was the Lesk Algorithm [Les86] (1986); which performed a disambiguation of two or more words by finding the pair of dictionary senses with the greatest word overlap in their dictionary definitions, since each definition was considered as a bag of words.

Nowadays, looking at the different corpora types employed, approaches to automatic disambiguation can be distinguished in: *knowledge-based methods*, which rely primarily on dictionaries, thesauri or semantic lexicons, without using any corpus evidence; *corpus-based or supervised methods*, that make use of sense-annotated corpora, manually constructed by disambiguating words in a corpus, in order to train a disambiguator; *unsupervised methods*, according to which, disambiguators work directly on unannotated corpora without the use of any external resource or training, and words' senses are individualized employing clustering.

An example of WSD's application can be the use of a lexical knowledge base such as WordNet in order to explore hierarchies and semantic similarity of words, similarly to Resnik's early work on WordNet ontology [Res95], to disambiguate analyzing similar words that are occuring in a same corpus/context.

Domain Adaptation and Target-Specific Polarity

"The accuracy of sentiment classification can be influenced by the domain of the items to which it is applied". A same word or phrase can indicate different sentiments in different domains [PL08].

The problem was pointed out before others by Turney in [Tur02] where he evidenced the domain-dependency of adjectives. Turney illustrated that an adjective such as "unpredictable" expresses a positive sentiment if it refers to a "movie plot" but negative if it is descripting a "car's steering abilities" [PL08].

As it is explained in [KFP09], it has to be noted that the problem of domain-dependency *"has nothing to do with word sense ambiguity. Even if the word sense is identified, the polarity still might be open"*. In fact, for example, in both the previous cases, "unpredictable" adhere to WordNet word-sense 1: "not capable of being foretold".

Moreover, as it is argued in [FK08], even within a domain, the polarity of adjectives can vary. To determine correct sentiment polarities is not enough to recognize the right domain; semantic orientation often depends on the target entity to which a sentiment refer, also in the same domain. For example, considering a "food and drinks" domain, "cold burger" and "cold pizza" are negative expressions, while "cold beer" and "cold coke" are positives; the same for "old wine" (positive) as compared to "old bread" (negative). In the Opinion Mining field a polarity of an adjective depending on the accompanying noun is called *target-specific polarity*; to sum up, a word might take both polarities, positive and negative depending on the domain-specific target object.

Dependency Analysis and Coreference Resolution

Coreference resolution is an activity that has been studied extensively in computational linguistics (i.e. Natural Language Processing)[Liu10b]; it consists in the correct interpretation of which is the referent of a linguistic expression (sentence) within a discourse; if it is applied in Opinion Mining, it has the main purpose of the understanding to who (or which entity) a sentiment refers; therefore, it can be useful to solve target-specific polarity.

Taking a text excerpt as example such as "Domenico gave me the camera that he bought", it should be easily infered that "that" refers "camera". But, as it similarly happens for word senses, co-references can be ambiguos; although "he" seems to refer to "Domenico", it could refer to someone else introduced earlier in the discourse [Liu10b].

Parsing is an operation employed by several researchers in order to analyze syntactic dependencies occuring between words, making use of generated dependency (or syntactic parse) trees.

"Studies have shown that effectively applying the technique to sentiment analysis can improve classification accuracy by about 10%" [NSI08].

Coreference Resolution, which represents a task easy for humans, it is still a major challenge in automatation [Liu10b].

Conclusions

We decided to employ *SentiWordNet* in our Sentiment Classification in order to have weights at analysis disposal.

In the calculation of reviews' scores are involved *prior polarities* of *individual words* within a review; the collection of the terms for the score calculation is limited to the words matching the four *part-of-speech* tags (adjectives, adverbs, nouns, verbs) by which SentiWordNet is partitioned; moreover, only the meaning of the words that match their part-of-speech tags should be considered to calculate the sentiment, the others should be discarded because no meaningful. Therefore a *tokenization* and a *part-of-speech tagging* processes are necessary, knowing that both can effect subsequent processing.

We decided that it is actually out of scope to use any sophisticated *target-specific polarity* or *coreference resolution* for this project.

Also *Word Sense Disambiguation* is not involved in this work; however we decide to experiment with different methods for the selection of the word sense in order to understand which benefit there could be in a future usage of such a feature.

Being aware that *polarity shifters* can affect classification accuracy, we take into consideration the *detection* of *negations* and other *polarity expressions*.

Chapter 5

Design

In this chapter, features of *Classifier* and structure of *Database* and *System Architecture* are presented. It is also described the *refinement* of them through the main stages of the work. Moreover, a *Word Sense Disambiguation study* is reported.

5.1 Rule-Based Classifier and Approach Investigated

In the social networks considered, Yelp and Foursquare, users can post opinions about some locations, such as bars, pubs and restaurants.

The **Rule-Based Classifier** has the main functionality to classify the reviews collected from the social networks by their overall sentiment (semantic orientation or polarity) in positive and negative.

At the moment it is chosen to build and evaluate **one Baseline** and **one** more functional **Enhanced Classifier**. The idea is to follow an incremental development and evaluation; where more modules and rules can be added to the initial baseline, in order to improve the effectiveness of the system.

The *classifier task* is to compute the overall sentiment of the reviews looking at the prior polarity of the individual terms contained in it (unigrams). Basic *Natural Language pre-Processing techniques* are applied together with *rules*. While *domain* is not taken into account, *context* is partially analyzed in the refinement part of this work; where for context of a word we mean other lexical items surrounding it; and both, domain and context, can influence a word attitude. Consequent to the previous concept, polarity of words can be prior or contextual.

In this project we start considering prior polarity (*Prior-Polarity Classification*) with the intent of further improving performance by switching to the contextual one (*Contextual Polarity Classification*); to accomplish that task, it is necessary to look at relations between words and optionally at group of subsequent words (e.g., bigrams or trigrams), moving towards the *Rule-Based Approach*.

As we have already seen in Section 4.3.2, different lexicons can be used to determine the semantic orientation of a word; there are subjectivity lexicons, such as the one introduced (by Wilson et al.) in [WWH05], that just list words that are considered subjectives; and other lexicons, such as SentiWordNet [BES10], that assign them a score representing their polarity strength.

We decided to employ SentiWordNet in our Sentiment Classification in order to have weights at analysis disposal and take advantage of a granularity in terms of sentiment reviews' scores that may be capitalize in a future reviews' ranking process; where reviews may be ranked by how positive/negative they are.

5.2 Baseline for Sentiment Classification

In this part of the work we describe the *Basic Sentiment Analysis* pipeline involved in the *Classification*.

5.2.1 Classifier Features and Pre-Processing Operations

In the *first Sentiment Classification* study, we have experimented with a *single word* or token *model* (unigram model, if compared to unsupervised methods); all the individual words appearing in reviews' text and matching some pre-selected *part-of-speechs*, were considered, as sentiment features, being involved in the classification task; although, we are aware that such a model is not sufficient for accurate sentiment classification, since words can change their sentiment combined with others. A more correct rule-based approach should involve: the application of a generated set of patterns; the analysis of context and domain, where both can influence a word attitude; the complementary employ of set of consecutive words (e.g., n-grams such as bigrams) to better capture patterns of sentiment expressions.

We used tokenization and part-of-speech tagging to select unigrams.

Tokenization consists in splitting text into pieces called tokens. We approach tokenization in stages, starting with sentences and moving on tokens. Sentences may be helpful in a future analysis process to resolve problems such as coreference. Word tokenization can involve the splitting of the text everytime a punctuation mark is reached or a space is individualized (white or blank space tokenizer). Anyway, tokenization based on whitespace is inadequate for many applications because it brings punctuation together with words. Moreover, tokenization involve issues in determining how to correctly tokenize; for instance [MRS08], considering "for aren't", it should be used a single token, such as "aren't" or "arent", or two tokens "aren" and "t"? Since tokenization decisions can effect part-of-speech tagging and other subsequent processing, we decided to use a *regular expression tokenizer*; regular expressions are a powerful and flexible method of specifying patterns; consequently, text is divided in substrings following the generated-specified patterns, where better tokenization can derive from a more complete pattern specification.

Each token is normalized using **normalization** rules for the English language; *"token normalization is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens"* [MRS08]. *Short forms' expansion* is employed to eliminate contractions; for example, the word "aren't" is replaced with "are not" or "it's" is replaced with "it is". A set of contraction rules is compiled looking at natural language, to the purpose. Terms are also transformed to *lowercase* in order to avoid problems during search for entries in the SentiWordNet database; for the same previous reason, words are brought to their base form through *lemmatization*; lemmatization is a technique that consists in removing inflection endings from a word in order to carry back its dictionary form also known as lemma; considering "am", "is", "are", for example, the lemma is always "be" [MRS08].

The individuation of a token's role within a discourse implies **part-of-speech tagging**. Part-of-speech tagging is used in this work to identify words, corresponding to part-of-speechs, that are good predictors of sentiment in sentences.

"Part-of-speech (POS) information is commonly exploited in sentiment analysis and opinion mining. One simple reason holds for general textual analysis, not just opinion mining: part-of-speech tagging can be considered to be a crude form of word sense disambiguation" [PL08].

Part-of-speech tagging can be based on different methods, described in [JM07] as following:

1. Rule-Based taggers: they involve *"a large database of hand-written disambiguation rules; for instance, a rule can specify that an ambiguous word is a noun rather than a verb if it follows a determiner"*.

2. Stochastic taggers: *"they generally resolve ambiguities by using a training corpus to compute the probability of a given word having a given tag in a given context"*

   a. N-grams taggers: they are trained using a tagged corpus to determine which tags are most common for each word, given adjacent part-of-speechs;

   b. Probabilistic methods:

      - HMM (Hidden Markov Model) based tagging: it assigns the most probable tag given N previous tags; it is a special case of Bayesian inference or Bayesian classification (eg TnT – a statistical part-of-speech tagger);
      - Maximum Entropy Model tagging.

3. Transformation-Based tagger (TBL) (e.g., Brill tagger); it shares features of both tagging architectures. *"Like the rule-based tagger, it is based on*

rules which determine when an ambiguous word should have a given tag. Like the stochastic taggers, it has a machine-learning component where the rules are automatically induced from a previously tagged training corpus".

In this study, we used an already implemented version of the Treebank POS tagger, based on on the maximum entropy model.

We limited the collection of terms for a review's score calculation to the four p-o-s tags (adjectives, adverbs, nouns, verbs) corresponding to the SentiWorNet partitioning. Only the meaning of the words that match those tags are considered to calculate the sentiment, the others are discarded because no meaningful.

In the future, the part-of-speech module may be have the role to select part-of-speechs to which apply pre-specified patterns.

5.2.2 SentiWordNet

SentiWordNet is *"a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications"* [BES10]. Version 3.0 is available substituting the previous version 1.0. SentiWordNet is based on WordNet (version 3.0), as it is easily understandable from its name.

WordNet is a semantic database for English language developed by Princeton University; its purpose is to organize, define and describe the concepts expressed by the terms contained in it. WordNet is a linguistic resource, a dictionary, in which the terms are related to each other forming an usefull *word's network.*

In WordNet terms are divided in four main categories:

- nouns
- verbs
- adjective
- adverbs

Terms are linked by meaning and grouped in *synsets* (*synonym sets*).

The entity *synset* groups every word having the same meaning; for example, words as "subject" and "topic" belong to the same *synset*; viceversa, "topic" can be described through other words belonging to the same *synset*, like "subject" and "theme".

There are different types of relationships in WordNet, such as *synonyms* and *antonyms*, or *hypernyms* and *hyponyms*, where *hypernyms* is a word generalization and *hyponyms* is a word specialization.

SentiWordNet is *"the result of automatically annotating all Wordnet synsets according to their degrees of positivity, negativity, and neutrality"*; each *sysnet s* has three numerical scores *Pos(s), Neg(s), and Obj(s)* ranging in the interval [0,1]; it is possible to have a positive, negative and objective score grater than zero at the same time; their sum is always 1 [BES10].

Each synset has always a gloss associated to it, as depicted in Figure 5.2.

SentiWordNet was constructed on the base of WordNet on the hypothesis that similar synsets have similar glosses. *"A gloss is composed by the list of the terms belonging to the synset, the concept definition and, optionally, some sample phrases"* [Esu08b].

Moreover, because of the polysemy of words in Natural Language, words can assume the role of different part-of-speechs in different contexts, and having different meanings, they can occur in more than one synset. In WordNet, and SentiWordNet as well, in order to distinguish various senses of a term, a sense number is associated to them. In the next Section 5.2.3, this concept will be described more in detail with an example.

Figure 5.1 shows the graphical model that has been designed by Esuli and Sebastiani [ES06] to display the scores of a synset in SentiWordNet.

"Esuli and Sebastiani expanded WordNet by adding polarity (Positive-Negative) and objectivity (Subjective-Objective) labels for each term. The resulting mapping is a two-dimensional representation of the word's emotional polarity and strength" [Mej10].

This model is used in the Web-based graphical user interface through which SentiWordNet can be freely accessed at `http://sentiwordnet.isti.cnr.it/`.

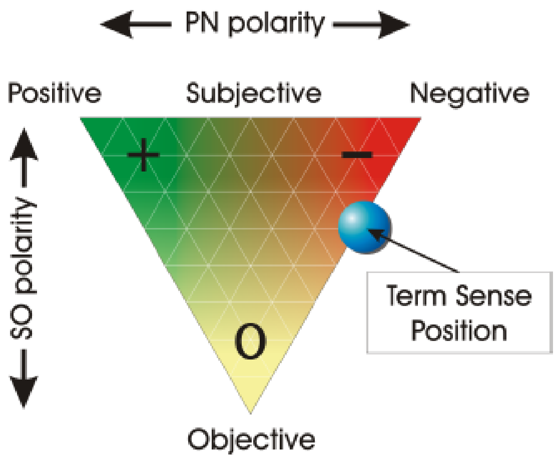

Figure 5.1: The graphical representation adopted by SentiWordNet for representing the opinion-related properties of a term sense [ES06].

5.2.3 Classification Algorithm

SentiWordNet

The classification algorithm takes in input words' tokens coming out from the pre-processing phase described in Section 5.2.1, together with relatives part-of-speech tags assigned.

The collection of the terms involved in the calculation of the review score is reduced to the ones for which it is possible to obtain a score; therefore, if a tokenize word belongs to one of the four p-o-s classes of SentiWordNet (adjectives, adverbs, nouns, verbs) it is looked up in SentiWordNet.

Token Scores' Triple

As it was explained in Section 5.2.2, words in Natural Language are polysemouses and because of multiple meanings tokens can have multiple entries in SentiWordNet. Consequently, in order to assign the polarity score to a word, it is first necessary to perform *Word Sense Disambiguation (WSD)*.

In this work no Word Sense Disambiguation is involved; it was decided

```
# SentiWordNet v3.0.0 (1 June 2010)
# Copyright 2010 ISTI-CNR.
# All right reserved.
#
# For any information about SentiWordNet:
# Web: http://sentiwordnet.isti.cnr.it
# -------
#
# Data format.
#
# SentiWordNet v3.0 is based on WordNet version 3.0.
# WordNet website: http://wordnet.princeton.edu/
#
# The pair (POS,ID) uniquely identifies a WordNet (3.0) synset.
# The values PosScore and NegScore are the positivity and negativity
# score assigned by SentiWordNet to the synset.
# The objectivity score can be calculated as:
# ObjScore = 1 - (PosScore + NegScore)
# SynsetTerms column reports the terms, with sense number, belonging
# to the synset (separated by spaces).
#
# -------
#
# POS   ID       PosScore        NegScore        SynsetTerms     Gloss
a       00001740        0.125   0       able#1  (usually followed by `to') having the necessary means or skill or k
a       00002098        0       0.75    unable#1        (usually followed by `to') not having the necessary means c
a       00002312        0       0       dorsal#2 abaxial#1      facing away from the axis of an organ or organism;
a       00002527        0       0       ventral#2 adaxial#1     nearest to or facing toward the axis of an organ or
a       00002730        0       0       acroscopic#1    facing or on the side toward the apex
a       00002843        0       0       basiscopic#1    facing or on the side toward the base
a       00002956        0       0       abducting#1 abducent#1  especially of muscles; drawing away from the midlir
a       00003131        0       0       adductive#1 adducting#1 adducent#1      especially of muscles; bringing tog
a       00003356        0       0       nascent#1       being born or beginning; "the nascent chicks"; "a nascent i
a       00003553        0       0       emerging#2 emergent#2   coming into existence; "an emergent republic"
a       00003700        0.25    0       dissilient#1    bursting open with force, as do some ripe seed vessels
a       00003829        0.25    0       parturient#2    giving birth; "a parturient heifer"
a       00003939        0       0       dying#1 in or associated with the process of passing from life or ceasing t
a       00004171        0       0       moribund#2      being on the point of death; breathing your last; "a moribu
a       00004296        0       0       last#5  occurring at the time of death; "his last words"; "the last rites"
a       00004413        0       0       abridged#1      (used of texts) shortened by condensing or rewriting; "an a
a       00004615        0       0       shortened#4 cut#3       with parts removed; "the drastically cut film"
a       00004723        0       0       half-length#2   abridged to half its original length
a       00004817        0       0       potted#3        (British informal) summarized or abridged; "a potted versic
a       00004980        0       0       unabridged#1    (used of texts) not shortened; "an unabridged novel"
a       00005107        0.5     0       uncut#7 full-length#2   complete; "the full-length play"
a       00005205        0.5     0       absolute#1      perfect or complete or pure; "absolute loyalty"; "absolute
a       00005473        0.75    0       direct#10       lacking compromising or mitigating elements; exact; "the di
```

Figure 5.2: SentiWordNet's database (excerpt).

to postpone it to a further classifier.

Since each word in SentiWordNet has multiple senses, each pair word-sense is collected together with the three corresponding polarity scores: positive, negative and objective.

Then, we applied and evaluated three different *strategies* for the calculation of the final *triple* of *Token Scores*, in order to understand which benefit there could be in a future usage of a Word Sense Disambiguation approach:

- *Random Sense*

- *All Senses Arithmetic Mean*

- *P-O-S matching Senses Arithmetic Mean*

The third of the previous methods is the one that should be most close to a WSD module's functionality/behaviour and should give best results in accuracy.

In constrast, the first is the simplest one that should achieve worst results. In fact, the method just consists in the *Random* selection of a *Sense* between all the ones collected for a term.

The second represents the *Arithmetic Mean* of each of the three polarity scores computed on *All* the possible *Senses*, an average of the sentiment entries of the word for all possible p-o-s taggings.

The *P-O-S matching Senses Arithmetic Mean* represents also an average of the sentiment entries of the word, but the entries to take into account are reduced to the ones that match the p-o-s tag assigned in the pre-processing phase. Therefore not all senses are considered but just the senses of the words in SentiWordNet that are matching the computed part-of-speech tag; if more than one sense belongs to the subset obtained after the p-o-s tagging filtering, then the arithmetic mean is applied.

Each of the three scores methods can be seen as a superclass of the three polarity classes: positive, negative, objective; therefore, in the algorithm computation, at the end of this step, for each *token* T we will have nine different scores:

- *Random Sense Pos Score (T)*
- *Random Sense Neg Score (T)*
- *Random Sense Obj Score (T)*
- *All Senses Arithmetic Mean Pos Score (T)*
- *All Senses Arithmetic Mean Neg Score (T)*
- *All Senses Arithmetic Mean Obj Score (T)*
- *P-O-S matching Senses Arithmetic Mean Pos Score (T)*
- *P-O-S matching Senses Arithmetic Mean Neg Score (T)*
- *P-O-S matching Senses Arithmetic Mean Obj Score (T)*

The last six scores are resulting from the formula:

$$score_{pol}(T) = \frac{1}{n} \sum_{s=1}^{n} score_{pol(s)} \tag{5.1}$$

where $pol \in \{pos, neg, obj\}$, and n is the number of the s *synsets* for the *token* T; n is reduced to a subset of all the synsets when it is considered the P-O-S matching.

In a previous moment it was also decided to evaluate the influence of applying weights to token scores as a function of the occurences of a term in a review. After some experiments, in which, to the previous formula, a multiplicative factor representing the *term frequency* of a token in a review was applied, the idea was discarded. Local weighting, in fact, does not work well in short documents and was resulting no meaningful in the case of the collected reviews.

Token Score

After analyzing the most appropriate kind of *Score* to select for a *Token*, between *Random Sense*, *All Senses Arithmetic Mean*, *P-O-S matching Senses Arithmetic Mean*, the final score triple of positive, negative and objective scores is obtained. At this time, the goal of the algorithm is to determine the *semantic orientation* of the *word* considered and its *Token Score*; the approach adopted is similar to the one reported in [DZC10] [Den09]: positive and negative SentiWordNet scores for a term are compared; if the positivity (or negativity) is larger, the word is considered positive (negative, respectively) and its strength is represented by its positivity (negativity, respectively) score. If both values are equal, the word is ignored, since the interest is toward opinionated words. The objective value is take into account in the case we want to apply a cut-off in order to exclude, from the computation of the overall sentiment review score, words that are too much objectives.

Normalized Review's Score

The formula to calculate the overall *Sentiment Score* of a review R consists in subtracting the sum of the scores of its negative words from the sum of its positive words' scores:

$$SentiScore(R) = \frac{\sum_{pos=0}^{j} Score(Token_{pos}) - \sum_{neg=0}^{k} Score(Token_{neg})}{j + k} \tag{5.2}$$

where j and k are the number of positive and negative words in R respectively, *Token* is a word in R, *Score(Token)* is the highest SentiWordNet score of the word among the couple positive-negative.

Since if the review is longer, it can contain more words that can be positive or negative, more high or low may be the sentiment score, *SentiScore* is divided by the number of sentiment words in R, with the intent to dampen the affect of review's length on its score. With this step called **normalization** values are kept within the interval [-1,+1].

If *SentiScore(R)* is higher (lower, respectively) than zero then R is labeled as positive (negative, respectively); when *SentiScore(R)* equals to zero, it means that the score of positive words equals the score of negative words, then the review is considered objective.

A problem for the rule-based approach is to decide for a polarity value or range determining the classification. Taking a look to other papers, in

[MSPN10] an empirical study was conducted with the intent of establishing *"a sentiment range for determining whether a review R should be treated as positive, negative, or neutral"*; a further work may be to similarly evaluate a "classification range", with a training phase, that should help to achieve a more precise classification.

Methods

On the *SentiScore* formula, we decided to apply several methods (or classifier models) in order to investigate in the experiments which is the best way to apply for the classification procedure.

The first consists in the choice of **considering**, **or not**, **nouns** in the estimation of the *SentiScore*. At the beginning of this Section it was explained that the words in SentiWordNet are partitioned in adjectives, adverbs, nouns and verbs. Sometimes nouns are judged to be objectives words; and in some papers' experiments they are completely excluded.

The second consists in trying to apply a **cut-off** on the objective score of a token, in order to exclude, from the computation of the *SentiScore*, words that have a quite high degree of objectivity. It has to be noted that in SentiWordNet it is not excepted that a word can be contemporaneously positive, negative and objective; in fact, it is possible to have a positive, negative and objective score grater than zero at the same time; and most of SentiWordNet words have an objective score grater than zero, also if they are positives or negatives. We decided that the condition to be passed, to consider a word to be polar and to include it in the computation, has to be in the form of *"`ObjScore(T) < 1-(cutoff)`"*; since in SentiWordNet the summation of the positive, negative, objective scores for a term is 1, and the objective score results from the complement of positive plus negative scores, the cut-off applied can be considered to be slight. For example a cut-off of "0.3" will exclude words only if the objective score is higher than "0.7", that is still an high limit. With this kind of cut-off we expressly permit to include in the computation more words, polarity words that are also objectives, and maybe strongly objectives; anyway, our reviews are very short and an high cut-off, together with the condition on the p-o-s tag matching, may reduce words considered polar to a very small or empty set. For a word that passes the cut-off condition, the algorithm then apply the usual procedure comparing its positive and negative scores, and subsequently the *SentiScore* formula.

Token Score Algorithm

Next, we show an extract of the *algorithm* regarding the strategy *to compute the semantic orientation of a word*; where POS can be restricted to just {verbs, adverbs, adjectives} in the case it is chosen to not consider noun p-o-s senses.

If positivity and negativity values are equal, the subtraction between the polarity scores will give the result of zero; therefore there is no sense to include the word in the computation of the *SentiScore* formula; again, the word is ignored, since the interest is toward opinionated words.

```
for each Token = POS
consider the Score Triple calculated using the chosen score Method

  if ObjScore(T) > 1-(cutoff):
   do not include the word in the SentiScore computation

  else
   if PosScore(T) > NegScore(T):
   add Token,Scores(Token) to positive set

   if NegScore(T) > PosScore(T):
   add Token,Scores(Token) to negative set

   if PosScore(T) = NegScore(T):
   do not include the word in the SentiScore computation

end for each
```

Example

To figure out how the algorithm works we will give an explanation through an example.

Let's suppose to have a review and a set of extrapolated tokens in which the word "scream" appears. The word "scream" has 6 entries (synsets) in SentiWordNet; for 3 times it is labeled as a noun and for 3 times as a verb. For both the part-of-speechs it assumes the form of "word#sense number", like for example "scream#1", "scream#2", "scream#3". Let's assume that the p-o-s tag given to our word in the review in noun.

If the *Token Score* method is *Random Sense*, it will be chosen a random score triple between the 6 entries, ignoring the p-o-s tag given in the pre-processing phase; similarly, if the *Token Score* method is *All Senses Arithmetic Mean*, the final token score triple will be the result of the average executed on all the 6 entries; on the contrary, if the *Token Score* method is *P-O-S matching Senses Arithmetic Mean*, the average will be computed on the 3 triples corresponding to a noun P-O-S sense.

Let's consider the *P-O-S matching Senses Arithmetic Mean* and the noun P-O-S sense entries:

▷ scream#3, PosScore = 0.25, NegScore = 0.375, ObjScore = 0.375

▷ scream#1, PosScore = 0.125, NegScore = 0.0, ObjScore = 0.875

▷ scream#2, PosScore = 0.0, NegScore = 0.0, ObjScore = 1.0

After to have computed the *Token Scores* with the *Score Method* chosen, an unique score triple for the word "scream" will be obtained.

▷ scream, PosScore = 0.125, NegScore = 0.125, ObjScore = 0.75

If it is decided to not include nouns in the computation of the review score, the word "scream" will be discarded a priori because of its noun POS sense; otherwise, the polarity score triple will be then used to determine the semantic orientation of the word. In this last case if a cut-off is applied, for example of "0.3", the condition of *"`ObjScore(T) < 1-(cutoff)`"* is equal to true, because "0.75" > "0.7", and the word is excluded from the computation of the review score. If no cut-off is executed, *PosScore(T)* and *NegScore(T)* are compared and being equal the word is discarded too.

Literature

Looking at the literature, the most similar algorithm to our approach is presented in [Den08], where the overall sentiment score is performed applying *"a classification rule according to which each document whose positivity score is larger than or equal to the negativity score is classified as positive. Otherwise it is considered negative"*; a document in the case can be compared to my review. Rather in [Den09] *"to determine the polarity of a complete document, the number of positive, negative and objective words is calculated. If the number of positive words is larger than the number of negative words, the document is considered positive and vice versa"*. In both, the strategy for the calculation of the token scores consists in the arithmetic mean executed on the triple scores for all the term's senses found.

A cut-off approach is also applied in [Mej10] and [DZC10].

5.2.4 Database

An entity-relationship (ER) diagram or E-R model is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information: boxes are commonly used to represent entities; diamonds are normally used to represent relationships; and ellipses are used to represent entities' attributes, where an attribute is underlined if it is a primary key. An entity is an object or concept about which you want to store information; relationships illustrate how two entities share information in the database structure; the key attribute is the unique one distinguishing characteristics of the entity.

The construction of the model started from the the analysis of the data to which the social networks' APIs grant the access. Yelp and Foursquare APIs allow the retrieval of locations and reviews information; a location is identified as "business" and as "venue" in Yelp and Foursquare respectively.

The gathering of reviews pass through the use of different APIs methods, which allow to search for a list of businesses or venues, for example near a specific area, specified by a city or latitude/longitude of a geo-point.

Businesses and venues share most of the same attributes which are represented with a bright colour in Figures 5.3 and 5.4. "Reviews" and "categories" were locations' attributes transformed in entities because of being lists of attributes.

The Entity Relationship Diagrams (ERDs) reported in 5.3 and 5.4, representing the conceptual (or semantic) data model, were at a later stage (i.e. logical design) mapped into a relational schema representing the logical structure of the database (logical data model): the Extended Entity-Relationship (EER) diagram; it consists in the DBMS data model, represented by a collection of tables (logical design view), where data is described from a physical point of view specifying its structure (type, field size, etc).

The Extended Entity-Relationship (EER) model is generated with the "native" ER diagramming tool of MySQL: MySQL Workbench. Relationships are represented, using Crow's Foot Notation, as lines between the boxes; the ends of these lines are shaped to represent the cardinality of each relationship; attributes in white are optional while the ones in red are foreign keys.

A first version of the database was carried out in order to take trace of the reviews gathered from the social networks; for development purposes, cities and social networks were also treated as being entities; businesses and venues were joined in a single table having most of the attributes in common. A refinement of the database was after necessary for sentiment analysis and evaluation purposes, such as the storage of the tokens extracted from the reviews or the memorization of the sentiment reviews' scores. First and second versions of the realized database, corresponding to the opinion collection and opinion mining phases of this work, are shown in Figure 5.5 and Figure 5.6 respectively.

5.2.5 System Architecture

We approach the problem of Sentiment Classification in a Location-Based Social Networking Space in stages, starting with the *collection of reviews* from the social networks and moving on the *Sentiment Analysis and Classification.*

As depicted in Figure 5.7, reviews and other information contained in

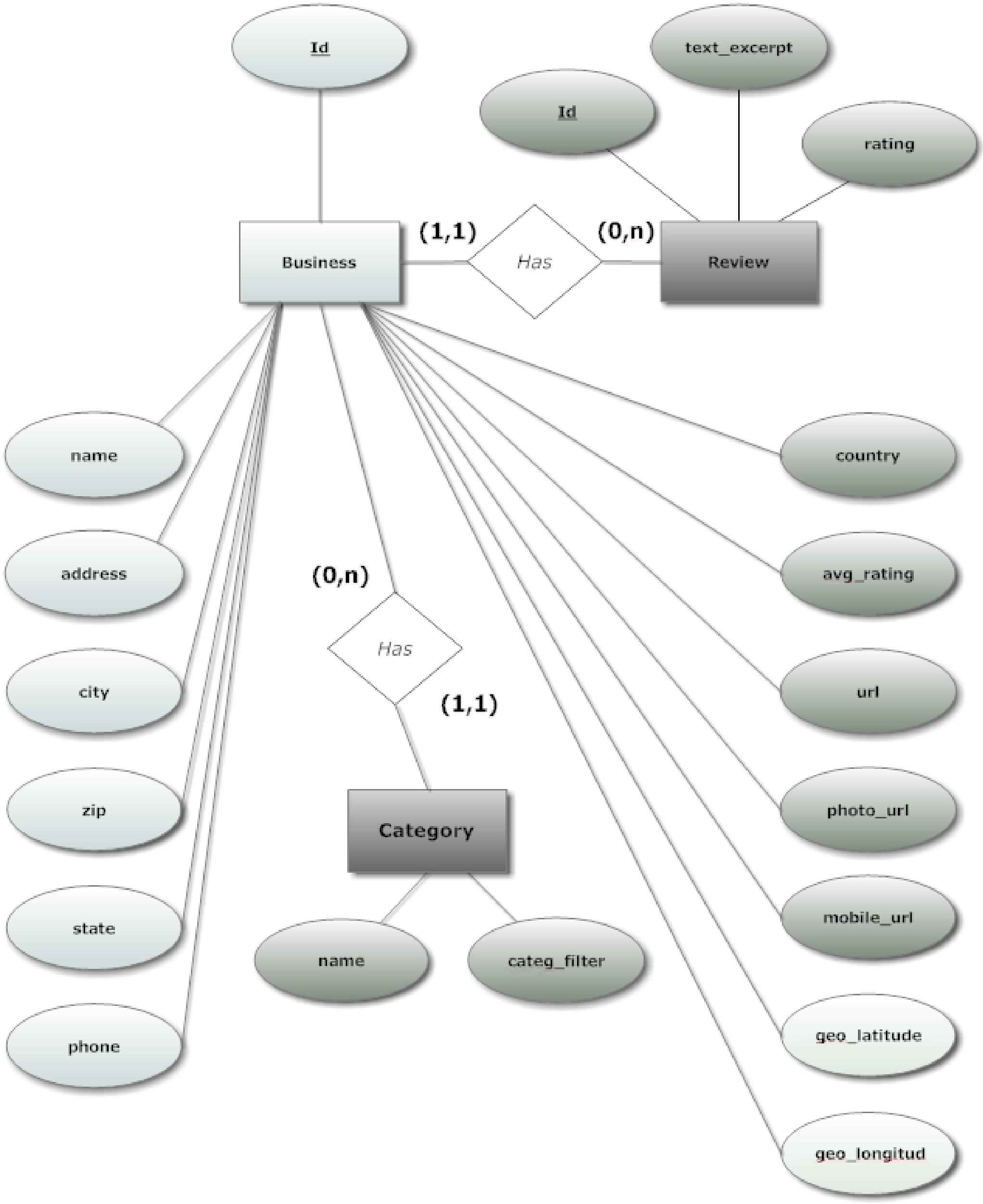

Figure 5.3: Yelp's Entity-Relationship Diagram.

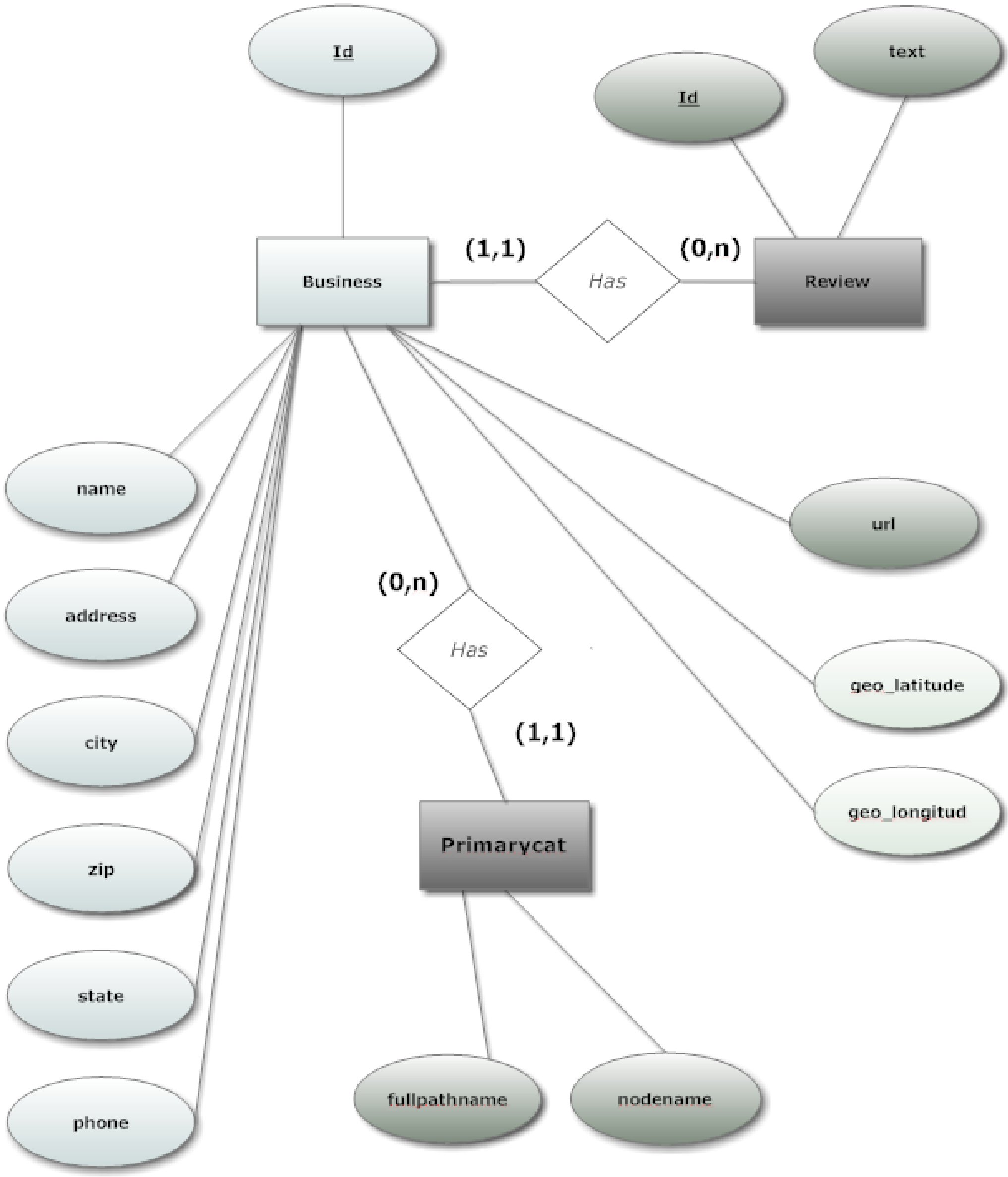

Figure 5.4: Foursquare's Entity-Relationship Diagram.

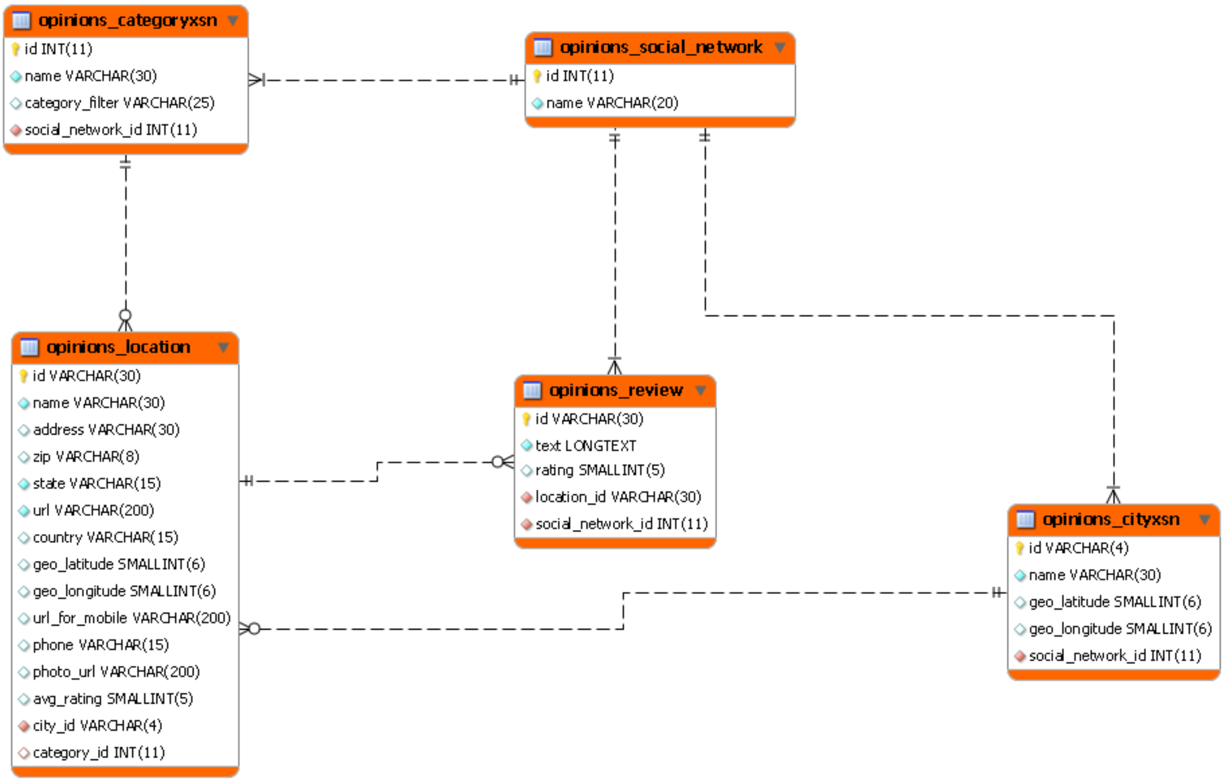

Figure 5.5: EER Diagram 1st Database's Version.

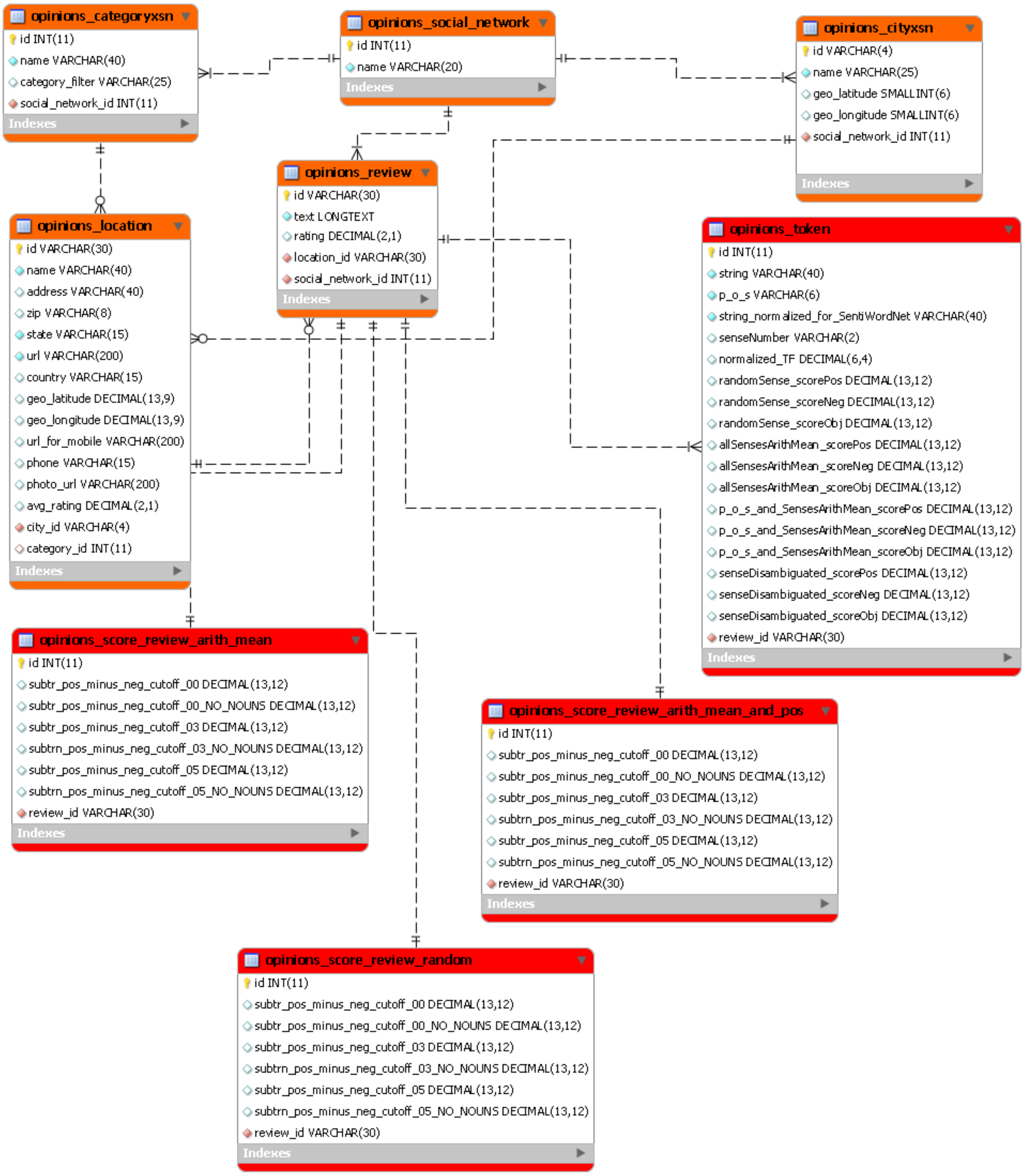

Figure 5.6: EER Diagram 2nd Database's Version.

Yelp's and *Foursquare*'s *repositories* are accessed through *requestes* to the *Web Services APIs* and returned by the *service providers* through *responses*. The obtained data are saved into the *database*.

In order to proceed with Opinion Mining, *reviews* are processed with several Natural Language Processing steps, as described in Section 5.2.1. *Tokens* (individual words) are generated one at a time; they are *normalized* and *tagged*, to be looked up in the *SentiWordNet* lexicon; if a lexicon entry corresponding to the token is found, the *token score algorithm* is applied and the resulting *token score* is sended to the *Prior-Polarity Classifier* to take part in the calculation of the *review's Sentiment Score*. *Token score algorithm* and *SentiScore formula* follow the classifier model selected during experimentation.

5.3 Refinement of the Sentiment Classification

In this second part of the work the *tokenization* process is refined, a module for *spelling correction* is designed, the identification of some *"slang expressions", emoticons, exclamations*, and the *detection of negations* are included in the Sentiment Analysis.

5.3.1 Enhanced Classification Algorithm: Refinement Steps

In this second part of the work the classification algorithm was refined; new features were included and others were extended.

The classification procedure always starts with **sentence splitting** and **tokenization**, which are modified in order to take into account emoticons and exclamations in both the processes. The algorithm tokenizes and splits in the case some **positive/negative emoticons** (e.g., ":))", ":@") or **exclamations** are found in text by a regular expression tokenizer. In order to evaluate the impact of more particular-nonstandard expressions in the sentiment analysis, it was decided to examine slang as well. A list of recognized **positive/negative slang expressions** (e.g., "thumbs up", "damnit") plus

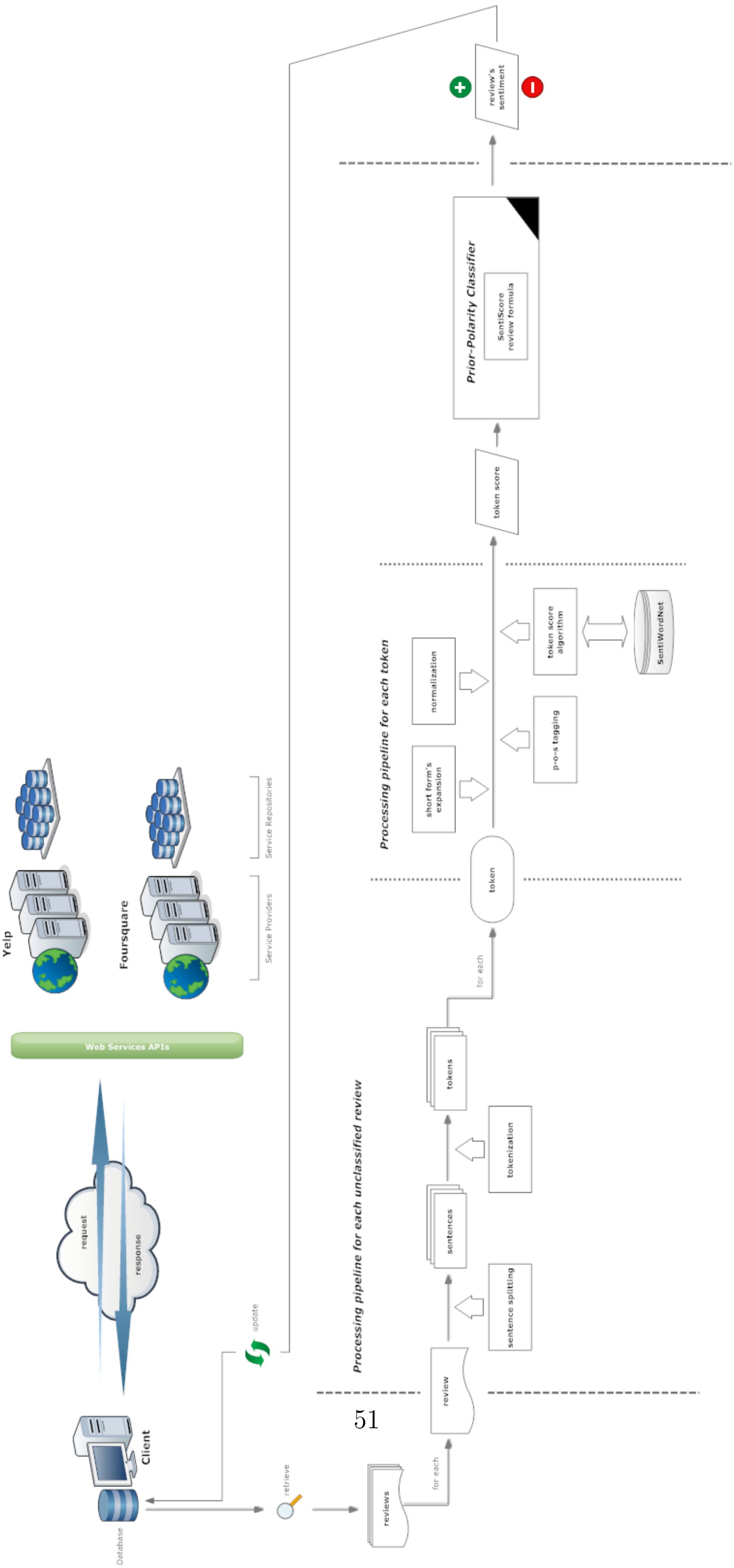

Figure 5.7: System Architecture and Sentiment Analysis Pipeline.

some others extracted from the “yes”, “no” categories of the General Inquirer lexicon was then compiled.

A function for **spelling correction** had also been designed. The tokens being normalized, as described in Section 5.2.1, are passed to a new module where their “correct” form is checked. The spelling corrector deletes repeated letters above twice (e.g., “goooood” ⇒ “good”) and identifies missing letters or extra letters occuring twice (e.g., “carot” ⇒ “carrot”, “niice” ⇒ “nice”); substitution of letters are made following few rules; letters are replaced and words are subsequently compared with english terms referring to corpora-dictionaries.

New features were inspired by the papers [TBP+10] and [PGS+10], where it is explained as people, posting text online and in social network sites, are usually ignoring the rules of grammar and spelling; as they write using abbreviations, slang, emoticons, *“repeated letters or punctuation for emphasis (e.g., a loooong time, Hi!!!)”* which can be *“reasonably effective in conveying emotion”*.

The classification algorithm was furthermore enriched with **negation detection**; it consists in identifying words which can reverse the sentiment orientation of surrounding terms.

In English, *negation* can occur in different ways, and it is tricky to predict and to handle with all negation cases. A *negation* usually inverts emotions of its subsequent terms, but it can also negate the concepts preceding itself (e.g., “Yesterday, at the disco, the good music and the nice people were absent”) or can effects sentences but just partially (e.g., “the music was not good at all but the atmosphere was amazing”). *Negation may be local (e.g., not good), or involve longer-distance dependencies such as the negation of the proposition (e.g., does not look very good) or the negation of the subject (e.g., no one thinks that it's good)* [WWH09]. More difficulties in modeling negation stands also in *conditional phrases* (if-clauses), *sarcasm* and *irony* [PL08] (e.g., “You could not play one piece correctly if you had two assistants”).

Predicting the *correct sentiment* of our reviews therefore can not only rely on term orientation but also on the *relations between terms within the* *context.*

In order to detect negations different techniques could be employed: *syntactic analysis* or *parsing*, for which it is necessary to declare a specific grammar relative to the text corpus, and related to which a *dependency analysis* can be performed in order to examine parent and child nodes of the negating word in the generated parse tree; *synctactic patterns*, like, for example, *part-of-speech tags* which can be used in combination with bigrams, trigrams or other n-grams in order to extract negation phrases (e.g., *"<verb>-Not-<verb>", "<verb>-Not-<adverb>-<adjective>"*) [NKW05] [NSK+04]; *regular expressions* and other *linguistic processing.*

Rules should be applied next in order to deduce an implied opinion from a detected expression.

In **[DC01] [PLV02]** negation detection is modelled by adding the tag "NOT_" to every word between a negation word and the first punctuation mark following. Several other methods for *"negation status identification"* have been developed in the recent years; most of them have been inspired by the **NegEx algorithm [CBH+01]** that **Chapman and colleagues** developed to find negated findings and diseases in medical records [GSNZ06]. The NegEx algorithm applies a regular-expression based approach which checks sentences for a match against a list of *"post-negations"* and *"pre-negations"*, in order to negate tokens, within a window of five words, preceding and following a negation term; double negatives, such as "not ruled out", or ambiguous phrasing, such as "unremarkable" are recognized as *"pseudo-negations"* and are therefore excluded from the negation identification [CBH+01]. However, the algorithm has been designed for the medical care field and it cannot be suited for our sentiment analysis purpose.

We decided to create **our own negation detection algorithm**.

We divided negations in *simple negatives* (e.g., "no", "neither", "never") and *connectors* (e.g., "but", "nor", "versus", "however"); lists of terms have been compiled looking at the *NegEx algorithm*, the *General Inquirer lexicon* (i.e. "Negate" and "Negativ" categories) and other *Internet* sources. Since connectors have the role of linking phrases of opposite polarity, we decided, given a sentence, to derive *sub-sentences* when a connector is found (e.g., "I do not like Rolling Stones but Beatles which are my favourites" ⇒ "I do not like Rolling Stones", "Beatles which are my favourites").

Each *sub-sentence* is parsed to check for *negations*; if a negation is found, all the subsequent tokens are selected as possible terms to be negated; in order to be definitely negated they have to pass a comparison with an *exception list* (e.g., "no doubt", "not only"); moreover it has not to be present a *second negation* between the first negation and the end of the subsentence (e.g., "I do NOT always REGRET what I have done").

Since, as we reported before, negation can be local or involve long dependencies, we treated *commas*, other than *connectors* and *punctuations*, to invidualize the context window of which tokens has to be negated. *Commas* are treated in order to differently identify the *end of a phrase* or a *comma-separated list*; if between the negation term and a comma there are more words than a predefined threshold (*windowsize*), terms after the comma are not considered; viceversa, we are in the case of a list that has to be negated (e.g., "I was at the restaurant; I did not like the service, atmosphere, all in general" vs "I do not like when the restaurant is crowded, therefore we decided to change place and it was amazing").

It has to be remarked that o*ur method employes relative simple linguistic processing and does not cover all the cases of natural language negation, as well as not all the possible words that can negate a sentiment.* Moreover, the influence of the negation detection on other related works on sentiment classification has been registered as not more high than 10% in improvement of results.

Token Score Algorithm

The Enhanced Classification Algorithm takes advantage of the additional, linguistically-driven functionalities previously described, to point out the final reviews' sentiment orientation.

The formula 5.2 used to calculate the overall *Review's Sentiment Score* holds steady. The main difference with the baseline stands in the **Token Score** calculation; the *Token Score Triple* just refers to the **Token Score Method** *P-O-S matching Senses Arithmetic Mean*; moreover it is decided to **consider noun p-o-s senses** and to **not apply a cut-off**.

Since in SentiWordNet the polarity values can vary within the unit interval [0,1], we decided to assign to any positive/negative *emoticon, slang and exclamation* expression found in text, a positive/negative value of 1. In the case of *multiword slang expressions* the value is distributed among the tokens which constitute the expression; for example, considering the positive slang expression "damn fine", the tokens "damn" and "fine" will assume a value of $1/2 = 0.5$ both. An *exclamation expression*, composed of one or more exclamation points, will be judged positive or negative depending on the sentiment of its preceding token.

```
for each Token = P-O-S {'ADJ','ADV','NOUN','VERB'} and NOT belonging to a Polarity Expression
  consider the Score Triple calculated using the score Method ''P-O-S, AM''

      if PosScore(T) > NegScore(T):
        if T is not negated:
            add Token,Scores(Token) to positive set
        else:
            add Token,Scores(Token) to negative set

      if NegScore(T) > PosScore(T):
        if T is not negated:
            add Token,Scores(Token) to negative set
        else:
            add Token,Scores(Token) to positive set

      if PosScore(T) = NegScore(T):
            do not include the word in the SentiScore computation

end for each



for each Token belonging to a Polarity Expression
  consider its pre-assigned Polarity Score

    if Polarity Score(T) > 0:
        add Token,PolarityScore(Token) to positive set
    else if Polarity Score(T) < 0:
        add Token,PolarityScore(Token) to negative set
```

```
end for each



for each Token = P-O-S {'EMOT_POS'}
    add Token,Score(1) to positive set
end for each



for each Token = P-O-S {'EMOT_NEG'}
    add Token,Score(1) to negative set
end for each



for each Token = P-O-S {'EXCL'}
  consider its Precedent Token and its Score

    consider the pT's Score

    if pT's Score = Pos:
        if pT is not negated:
          add Token,Score(1) to positive set
        else:
          add Token,Score(1) to negative set

    else if pT's Score = Neg:
        if pT is not negated:
          add Token,Score(1) to negative set
        else:
          add Token,Score(1) to positive set

end for each
```

5.3.2 Parallel Study on WSD

The task of *Word Sense Disambiguation* has been demontrated as being relevant for Sentiment Classification. The advantages are given in the superiority of the results if the disambiguation is correct; at the same time, it is easy to fall into errors which can significantly affect the classification quality. *"This provides further motivation to study in depth this problem"* [MBMP10].

Word Sense Disambiguation requires an analysis of the *context* of the words (i.e., apart of the word itself, every term that occurs before or after it) in order to assign them to the correct sense. Another source of information which can contribute to the disambiguation are *external knowledge resources*, such as lexicons or dictionaries.

We came up with the **idea behind our approach** looking at WSD related works. We found an external knowledge resource called "**eXtended WordNet**" which is introduced in [NB07] and which represents a disambiguated sense inventory for the WordNet glosses' definitions. Several researchers perform a disambiguation based on glosses' definitions, where different measures are employed in order to estimate the similarity between those glosses' words. As it was described in Section 3.2.2 the Lesk method [Les86] is based on counting words' overlaps between dictionary definitions, respectively related to the ambiguous word and the terms within its context; alternatively, the Jaccard or Cosine similarity [RPB04] can be used if bag-of-words or vector representation are adopted to represent glosses' definitions. Other research works describe the use of WordNet Relatedness measures based on the similarity between WordNet synsets; in [PBP03] and [PBP], measures of relatedness are *"plagged"* into an *"adapted Lesk algorithm"* in place of gloss overlaps.

We decided for **a procedure** that has to be **gloss-centered** in order to take advantage of eXtended WordNet, and that could combine part of the best techniques described before: it should consist in execute a disambiguation based on the words contained in the *gloss' definition* of the ambiguous review's terms, *where words' senses are identified by eXtended WordNet*; content words of the gloss definitions, already disambiguated by eXtended WordNet, should be used in order to find *relations between terms* within a certain context, which may help to assign each *correct sense.*

We subsequently designed **two techniques**; the first one is an adaption of the algorithm presented in [NB07], having a different set-up in order to be compared with the second one that is a totally new approach and that has been conceived to take advantage of the terms relationships defined in WordNet and some related similarity measures, other than eXtended WordNet; they consist in finding **overlaps** and **relatedness** respectively, during a comparison between *glosses' definitions, disambiguated by eXtended WordNet*, of possible senses of ambiguous words. The two techniques are thoroughly described in Section 5.3.4, while some relatedness' measures are presented below.

Measures of Semantic Relatedness

In WordNet words are organized in synsets, where each sysnet is related to each other by semantic relations; and **verbs and nouns** are organized in **hierarchies**; several measures have been developed to estimate their semantic relatedness working well together with the WordNet structure.

As it is reported in [PBP03], it has to be noted that even if the words-terms "similarity" and "relatedness" are usually interchanged, the second one specifies a more general notion; terms can be related without being similar (e.g., antonyms); anyway relatedness measures in the WordNet context refer to the relations of hypernymy/hyponymy (IS-A relationship, e.g., "red" is a "color") *"between pairs of nouns or pairs of verbs, since the concept hierarchies in WordNet do not mix parts of speech"*.

Relatedness measures can be divided in **path-based** and **information content-based**; they take in input a couple of words/concepts and they produce in output a degree of relatedness.

Information content is a measure related to the specificity of a concept; more specific is a concept to a particular topic, higher its information content value is. Information content takes its origin from the intuition of Resnik who stated *"the more similar two words are, the more informative will be the most specific concept that subsumes them both"* [Res95]. *"Thus, "carving fork" has a high information content, while "entity" has low information content"* [PBP03]. Information content can be calculated using a sense-annotated corpus and taking into account the more frequent sense that the word assumes in it; alternatevely, if a sense-annoted text is not available, it can be estimated counting the frequency of the word in the text/corpus and dividing it by the number of possible senses related to that word [PBP]. Information content can be formally described as the probability of encountering an instance of a concept in the corpus, provided via a maximum-likelihood estimation:

$$IC(concept) = -log(P(concept)) \tag{5.3}$$

The most common similarity-relatedness metrics based on *information content* refer to **Resnik** (Equation 5.4), **Lin** (Equation 5.6), and **Jiang & Conrath** (Equation 5.5).

Path-based similarities relate on the WordNet is-a hierarchy; **Wu & Palmer** (Equation 5.7) and **Leacock & Chodorow** (Equation 5.8) measures, as well as **Path Length** (Equation 5.9) score, are examples of them.

In Equation 5.9 *length* represents the number of nodes along the shortest path between two concepts, that is normalized in Equation 5.8 by the maximum depth D of the taxonomy. In the rest of the Equations, *least common subsumer (LCS)* is the most specific concept that the two concepts have in common, or better the concept that subsumes both the concepts *"(i.e., are ancestors of)"* in any sense of them [Res95].

Resnik formulates its similarity measure as the Information Content of the Least Common Subsumer of both the concepts; Jiang and Conrath, and Lin *"extend Resnik's measure by scaling the common information content values by those of the individual concepts"* [BP03].

In Equation 5.7 *depth* stands for the distance to the root node of the hierarchy; depth of the two given concepts in the WordNet taxonomy, and of the least common subsumer (LCS).

$$Sim_{res} = IC(LCS(concept_1, concept_2)) \tag{5.4}$$

$$Sim_{jcn} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS(concept_1, concept_2))} \tag{5.5}$$

$$Sim_{lin} = \frac{2 * IC(LCS(concept_1, concept_2))}{IC(concept_1) + IC(concept_2)} \tag{5.6}$$

$$Sim_{wup} = \frac{2 * depth(LCS(concept_1, concept_2))}{depth(concept_1) + depth(concept_2)} \tag{5.7}$$

$$Sim_{lch} = -log\frac{length}{2 * D} \tag{5.8}$$

$$Sim_{path} = \frac{1}{length} \tag{5.9}$$

We conducted our disambiguation study using an NLTK implementation of *Wu & Palmer* and *Leacock & Chodorow* metrics. Althought it is possible to calculate information content using NLTK and corpus like SemCor, Penn Treebank, British National Corpus (BNC) provided in the package, we excluded the information-content based measures from the evaluation due to the high requirements in terms of computational process time.

5.3.3 Extended WordNet

Extended WordNet[1] is a project owned by the University of Texas at Dallas with the aim of increase the connectivity between Wordnet synsets, semantically disambiguating content words of all the glosses. Extended WordNet is then a *disambiguated sense inventory* resulting from annotating all the glosses' terms, contained in each synset's definition of the WordNet dictionary, with their corresponding sense number.

The database has been generated automatically through a parsing and tagging process; the output has been partially checked with automatic voting process and human intervention, and the quality of each disambiguation is marked with an adjective.

The available release (XWN 2.0-1.1) used in this project is based on WordNet 2.0 and organized in four different xml files (adj.xml, adv.xml, noun.xml,

[1] `http://xwn.hlt.utdallas.edu/`

verb.xml) corresponding to the four part-of-speech classes of SentiWordNet (adjectives, adverbs, nouns, verbs).

In Figure 5.8, an excerpt of a xml file is shown; for each *gloss* it is indicated the *synset identifier*, the *synonyms' set* and the *text* contained in the *gloss* of the synset (*definitions + examples* (examples are quoted)); the *word sense disambiguation of the gloss'* definition is reported within the tag *"wsd"*; an adjective is used to mark the *"quality"* of the assertions. *Parse trees* and *logic form transformations* of gloss' definitions are also reported.

adj.xml

```
<gloss pos="ADJ" synsetID="00002062">
  <synonymSet>unable, not_able</synonymSet>
 <text>
   (usually followed by `to') not having the necessary means or skill or know-how; "unable to get to town without a car"
 </text>
  <wsd>
      <punc>(</punc>
      <wf pos="RB" lemma="usually" quality="normal" wnsn="1" >usually</wf>
      <wf pos="VBN" lemma="follow" quality="normal" wnsn="1" >followed</wf>
      <wf pos="IN" >by</wf>
      <punc>`</punc>
      <wf pos="TO" >to</wf>
      <punc>'</punc>
      <punc>)</punc>
      <wf pos="RB" lemma="not" quality="normal" wnsn="1" >not</wf>
      <wf pos="VBG" lemma="have" quality="normal" wnsn="1" >having</wf>
      <wf pos="DT" >the</wf>
      <wf pos="JJ" lemma="necessary" quality="normal" wnsn="1" >necessary</wf>
      <wf pos="NNS" lemma="means" quality="silver" wnsn="1" >means</wf>
      <wf pos="CC" >or</wf>
      <wf pos="NN" lemma="skill" quality="silver" wnsn="1" >skill</wf>
      <wf pos="CC" >or</wf>
      <wf pos="WRB" >know-how</wf>
  </wsd>
<parse quality="GOLD">
(TOP (S (NP (JJ unable) )
        (VP (VBZ is)
            (NP (NP (NN something) )
                (VP (RB not) (VBG having)
                    (NP (DT the) (JJ necessary) (NNS means) (CC or) (NN skill) (CC or) (NN know-how) ) ) ) )
        (. .) ) )
</parse>
 <lft quality="SILVER">
 unable:JJ(x1) -> not:RB(e1) have:VB(e1, x1, x6) necessary:JJ(x6) means:NN(x2) skill:NN(x3) or:CC(x6, x2, x3, x4) know-h
 </lft>
</gloss>

<gloss pos="ADJ" synsetID="00002287">
  <synonymSet>abaxial, dorsal</synonymSet>
 <text>
```

Figure 5.8: Extended WordNet's database (excerpt).

Extended WordNet can be exploited as Knowledge Base for applications in the field of Information Retrieval and Computational Linguistics.

5.3.4 WSD Algorithm

The **Word Sense Disambiguation algorithm** is applied to reviews that have been pre-processed with operations described in Section 5.2.1. The reviews are tokenized and part-of-speech tags are assigned; lexicon entries corresponding to the tokens are found in SentiWordNet. Usually more than one lexicon entry is found because of the *polysemy of the words*. Therefore, a disambiguation is necessary in order to assign the correct sense to each term.

It has to be notified that the disambiguation procedure **disambiguates words in couple**; and that not all the words present in the review text are compared with each other; the previous assertion is related to the notion of "context" that will be explained later on.

In Section 5.3.2 we described as our disambiguation procedure is **gloss-centered**. The glosses selected as input for our algorithm are the ones related to the possible senses of the words to disambiguate; it is important to specify that the possible senses/synsets, that should be included in the disambiguation, come out from a *filtering phase* during which only the *senses matching* the pre-assigned *part-of-speech tags* are confirmed.

In Section 5.3.2 we also stated that **two different techniques** have been designed within the WSD algorithm: a first one that works similarly to the Lesk method on gloss *overlaps* and a second one that investigates the *relatedness* of the terms contained in the gloss. The second one requires a high amount/time of computation because of the WordNet hierarchy that everytime has to be parsed. Therefore we decided to assign a *small size to the "context"*.

When we speak about **context** we intend the *"text consisting of content words in some small window surrounding to the target word"* [KB08]; more in the specific, it consists in the terms preceding and succeeding the word, including the word itself; in this case the context is identified as being *local*; sentence, paragraph, topic or domain define a *global* context. In most of the WSD works [NB07] [BP02], it is reported as a window of five words (no more than two words to the left and two words to the right) should be sufficient for the disambiguation. Anyway, there are some exceptions; in [NB07], for example, the whole sentence is involved in the disambiguation, while in [BP02] the authors experimented with a context window of three words. We believe that more words in the context window (i.e. a profuse quantity, enough words in order to work well) can help to achieve better disambiguation results; having said this, we thought a **context window** of nine words, four words to the left and four words to the right, plus the word to disambiguate itself, could be a good compromise with respect to time computation and accuracy/effectiveness; if subsequent/precedent words are not enough-selectable (i.e., terms at the end/begin of a sentence) the context window is covered with precedent/subsequent words.

Let's state the disambiguation **procedure**.

We can define the word to disambiguate as *"target word"* and the surrounding words belonging to the context window as *"content words"*.

Considering couples of {*target word, content word*}, *every sense/gloss of the target word* is compared to *each sense/gloss of the content words*. In order to disambiguate the glosses, five different kind of "**gloss-bags**" (i.e. bag-of-words) are used. One of them stores all the terms (lemmas) composing the gloss definition. The other four of them refer to the p-o-s classes in which SentiWordNet is partitioned (adjectives, adverbs, nouns, verbs); thereby, each gloss word (lemma) together with its disambiguated sense is placed in a different gloss-bag depending on its part-of-speech tag.

Gloss-bags are constructed for every sense of the couple {*target word, content word*}.

In Figure 5.9 it is reported an example for the first part of the disambiguation procedure of the *target word* "painter" within the *context window* ".. landscape whose style and direction ..".

Let's suppose to consider the couple "(painter#1, landscape#2)".

▷ The eXtended WordNet entry for the gloss of the noun "painter" with sense 1 is:

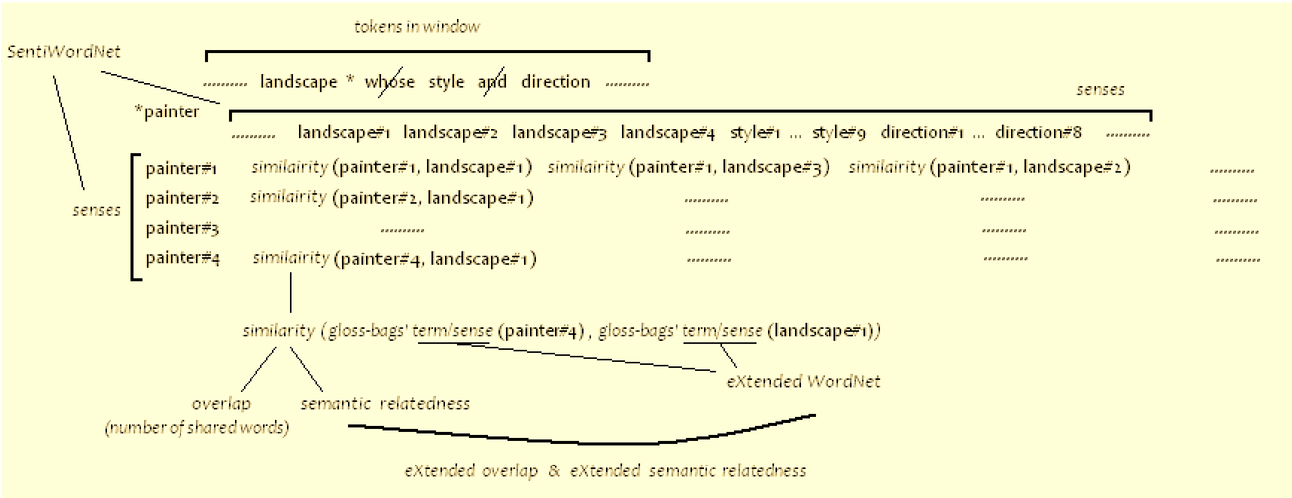

Figure 5.9: A Visual General Example for a Disambiguation procedure.

```
<gloss pos="NOUN" synsetID="09717847">
  <synonymSet>painter</synonymSet>
 <text>
   an artist who paints
 </text>
  <wsd>
      <wf pos="DT" >an</wf>
      <wf pos="NN" lemma="artist" quality="silver" wnsn="1" >artist</wf>
      <wf pos="WP" >who</wf>
      <wf pos="VBZ" lemma="paint" quality="normal" wnsn="1" >paints</wf>
  </wsd>
...
..
.
<gloss>
```

The generated gloss-bags are:

```
>>> glossBag
['*', 'artist', '*', 'paint']
>>> verbGlossBag
```

```
{'paint': '1'}
>>> nounGlossBag
{'artist': '1'}
>>> advGlossBag
{}
>>> adjGlossBag
{}
```

▷ The eXtended WordNet entry for the gloss of the noun “landscape” with sense 2 is:

```
<gloss pos="NOUN" synsetID="03504253">
  <synonymSet>landscape</synonymSet>
 <text>
   painting depicting an expanse of natural scenery
 </text>
  <wsd>
      <wf pos="NN" lemma="painting" quality="normal" wnsn="1" >painting</wf>
      <wf pos="VBG" lemma="depict" quality="normal" wnsn="1" >depicting</wf>
      <wf pos="DT" >an</wf>
      <wf pos="NN" lemma="expanse" quality="normal" wnsn="1" >expanse</wf>
      <wf pos="IN" >of</wf>
      <wf pos="JJ" lemma="natural" quality="normal" wnsn="1" >natural</wf>
      <wf pos="NN" lemma="scenery" quality="silver" wnsn="1" >scenery</wf>
  </wsd>
...
..
.
<gloss>
```

The generated gloss-bags are:

```
>>> glossBag
['painting', 'depicting', '*', 'expanse', '*', 'natural', 'scenery']
>>> verbGlossBag
```

```
{'depict': '1'}
>>> nounGlossBag
{'painting': '1', 'expanse': '1', 'scenery': '1'}
>>> advGlossBag
{}
>>> adjGlossBag
{'natural': '1'}
```

Once the gloss-bags are defined, the two different techniques of disambiguation come into play.

The *Extended Lesk* technique searches for overlaps, checking each lemma of a considered gloss-bag (e.g., noun gloss-bag) for the target word (i.e., "painter", with sense 1) with each lemma of the same gloss-bag for the content word (i.e., "landscape", with sense 2); an *overlap* is valid if the words match the same part-of-speech as well. The *score* relative to a match depends on the general gloss-bag vector that store all the words/lemmas of the gloss definition: if there is an overlap of N consecutive lemmas in the gloss definition, then the score assigned to the matching senses is equal to N; the score is 1 otherwise. The procedure is repeated for each couple {"sense_of_the_target_word", "sense_of_the_content_word"}, during which scores are taking place in a *"Lesk_score" vector*. The best match between senses comes out from the *maximum of the scores* stored in the vector "Lesk_score". At this point the target word has been disambiguated with the context word; then, *target word and the sense related to the best match* are memorized in another vector having the size of the context window: *"voting_target"*. Then the procedure starts again and it is repeated until the comparisons, between the target word and each each other content word presents in the context window, are completed. At the end of all the comparisons of all the senses of all the content words with all the senses of the target word, the *target word's sense that is confirmed more times in the "voting_target" vector is assumed as being the correct sense* (e.g., considering {painter#1, painter#2, painter#2, painter#1, painter#1, painter#3, painter#1, painter#1, painter#4}, sense

1 of painter is selected).

The *WordNet-based Relatedness* technique employes only the *"noun gloss-bag"* and the *"verb gloss-bag"* in the disambiguation procedure, since in WordNet, as it was described in Section 5.3.2, nouns and verbs are organized in concept hierarchies which do not mix parts of speech. *Wu & Palmer and Leacock & Chodorow metrics* are involved in the procedure, even if any other similarity measure can be used. The technique searches for *semantic relatedness*, checking each word/lemma's synset of a "noun gloss-bag"/"verb gloss-bag" for the target word (i.e., "painter", with sense 1) with each lemma's synset of the "noun gloss-bag"/"verb gloss-bag" for the content word (i.e., "landscape", with sense 2); where, in WordNet, synsets are uniquely identified by the triple: term, part-of-speech, term sense number.

The *score* relative to each *couple of synset/sense*, resulting from Equation 5.8 and Equation 5.7, are placed in two different vectors: *"LCH_score"* and *"WUP_score"*. Once every couple synset/sense has been analyzed, the *maximum of* the scores stored in the vectors *"LCH_score"* and *"WUP_score"* is selected. The two scores cannot be compared, since, differently from the "LCH_score", the "WUP_score" takes place in the unit interval; therefore, *target word and the sense related to the best match of "LCH_score" and "WUP_score" vectors are both memorized in the vector "voting_target"*, having this time the double size of the context window. At this point, the disambiguation proceeds as in the *Extended Lesk* technique. The procedure starts again and it is repeated until the comparisons, between the target word and each each other content word presents in the context window, are completed. At the end of all the comparisons, the *target word's sense that is confirmed more times in the "voting_target" vector is selected as correct sense.*

We can describe the *two techniques* with *Equation 5.10* and *Equation 5.11*; note that Equations refer to the disambiguation of only *one couple {target word, content word}*. We already stated that the context window consist in 2n + 1 words, n to the left of the target word, n to its right, plus the target word it self; if we denote each content word as w_i, where $-n \leq i \leq n$, then the target word can be referred to as w_0. Assume that each content word w_i has s_k possible senses ($1 \leq k \leq l$), and each target word w_0 has s_j possible senses ($1 \leq j \leq m$). Equations show that the algorithm computes a score for each sense of the *couple {target word, content word}*; the output is the sense of the target word that is most related to the other words in the window of context. Equation 5.11 differs from Equation 5.10 in the calculation of the maximun relatedness between senses/synsets glosses' terms instead of the longest overlap between them; synsets involved in the semantic relatedness calculation will always obtain a score during the comparison (differently from the overlap calculation); in order to not favor synsets/senses which have more nouns or verbs in their gloss-bags, the maximum is preffered to the summation.

$$Sense_{extOVERLAP}(w_0, w_i) = \max_{1<j<m} (\sum_{k=1}^{l} Overlap(s_{0,j}, s_{i,k})) \tag{5.10}$$

$$Sense_{extRELATEDNESS}(w_0, w_i) = \max_{1<j<m} (\max_{1<k<l} Relatedness(s_{0,j}, s_{i,k})) \tag{5.11}$$

$\triangleright$ **Algorithm: (I) Extended WordNet Glosses'Overlaps-Based Disambiguation**

```
for each Token_1 in a Review:
  consider the precedent/subsequent 2nd_Tokens within its window

  for each couple {Token_1, 2nd_Token}:
    consider every sense of Token_1 and every sense of 2nd_Token found in SentiWordNet,
    matching the part-of-speech assigned by the tagger

    for each selected sense of Token_1, sense of 2nd_Token:
     consider their definition's gloss bags
     (entire gloss-bag, adjectives' gloss-bag, adverbs' gloss-bag, nouns' gloss-bag, verbs' gloss-bag)
     built upon eXtended WordNet

     compute the OVERLAP betwen glosses' terms,
     where a term in the gloss has to have the same sense number in order to match

    end for each

    confirm the sense of the Token_1 that has the maximum OVERLAP
    between each other sense of the 2nd_Token
```

```
  end for each

  take the sense of the Token_1 that is confirmed more time in comparisons with
  2nd_Tokens within its window

end for each
```

▷ **Algorithm: (II) Extended WordNet-Based Relatedness Disambiguation**

```
for each Token_1 in a Review:
  consider the precedent/subsequent 2nd_Tokens within its window

  for each couple {Token_1, 2nd_Token}:
    consider every sense of Token_1 and every sense of 2nd_Token found in SentiWordNet,
    matching the part-of-speech assigned by the tagger

    for each selected sense of Token_1, sense of 2nd_Token:
     consider their definition's gloss bags
     (nouns' gloss-bag, verbs' gloss-bag)
     built upon eXtended WordNet

     compute the RELATEDNESS betwen synsets of the glosses' terms,
     where a synset is (uniquely) identified by the triple {term.p-o-s.sn}

    end for each

    confirm the sense of the Token_1 that has the maximum RELATEDNESS
    between each other sense of the 2nd_Token

  end for each

  take the sense of the Token_1 that is confirmed more time in comparisons with
  2nd_Tokens within its window

end for each
```

5.3.5 Database and Senti/Extended WordNet Mapping

The Extended Entity-Relationship (EER) model is generated with MySQL Workbench, as explained in Section 5.2.4; relationships and cardinality of each relationship are represented using the Crow's Foot Notation; attributes in white are optional while the ones in red are foreign keys.

Figure 5.10 represents in green the tables replacing the red tables of the previous version of the database, depicted in Figure 5.6, in order to support the new classification algorithm. In the current case, *Token* and *Review's Score* tables only refer to the best method resulting from the first part of this study: *POS matching Senses Arithmetic Mean Score* including *nouns*, *without cut-off*. Subsentences and slang expressions were taken into account during the negation phase and other refinement steps described in Section 5.2.1.

The additional version of the database carried out for the study on Word Sense Disambiguation is reported in Figure 5.11.

New sentences selected from the *"Semcor"* corpora were used for the test.

For every sense of every token, the *gloss-bags*, contaning the *words* belonging to the *SentiWordNet* synset's definition and their relatives *sense numbers* extracted from *eXtended WordNet*, were constructed and stored to be available to the disambiguation procedure.

In order to accomplish the previous task it was before necessary to perform the *mapping between SentiWordNet and eXtended WordNet*. Since the current available versions of eXtended WordNet and SentiWordNet are based on two different releases of WordNet (2.0 and 3.0 respectively), they differ on the synsets identifiers which cannot be used for the mapping between glosses. Therefore, it was decided to alternatively relate the two knowledge resources through synset's *synonym terms*, *gloss definition* and *gloss examples*; it has been calculated that the process works almost perfectly, against a not considerable mapping error of 3%. It is important to notify that every time a couple of concepts has to be disambiguated the two knowledge resources are involved and consequently the mapping comparisons take part in the disambiguation as well, significantly increasing the time of computation.

Apart of the mapping, the four xml files of eXtended WordNet, corresponding to the part-of-speech classes of SentiWordNet (adjectives, adverbs, nouns, verbs), were splitted in files of reduced dimensions in order to improve the scalability of the computation; some of them were quite big in size; in particular the file dedicated to noun senses stored 30.000.000 rows in 120 Megabytes.

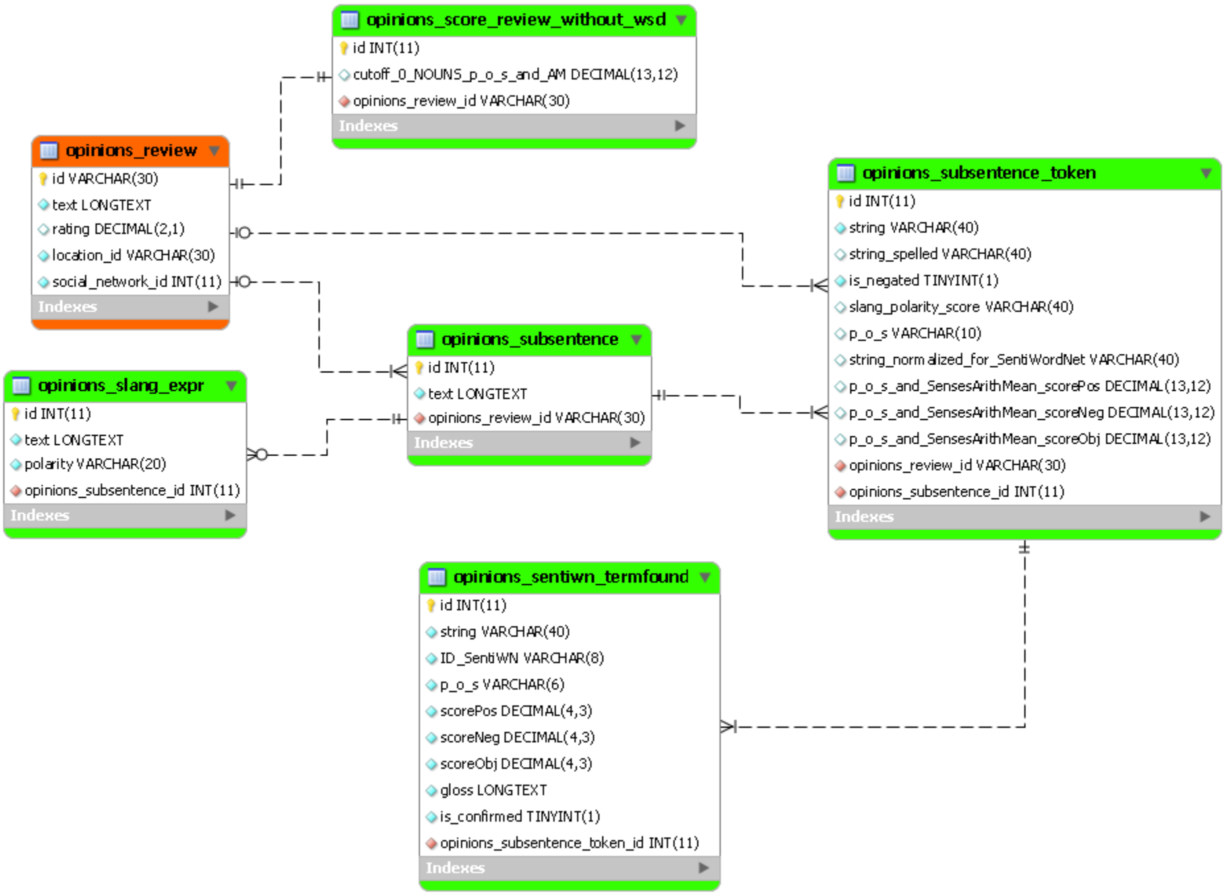

Figure 5.10: EER Diagram 3rd Database's Version.

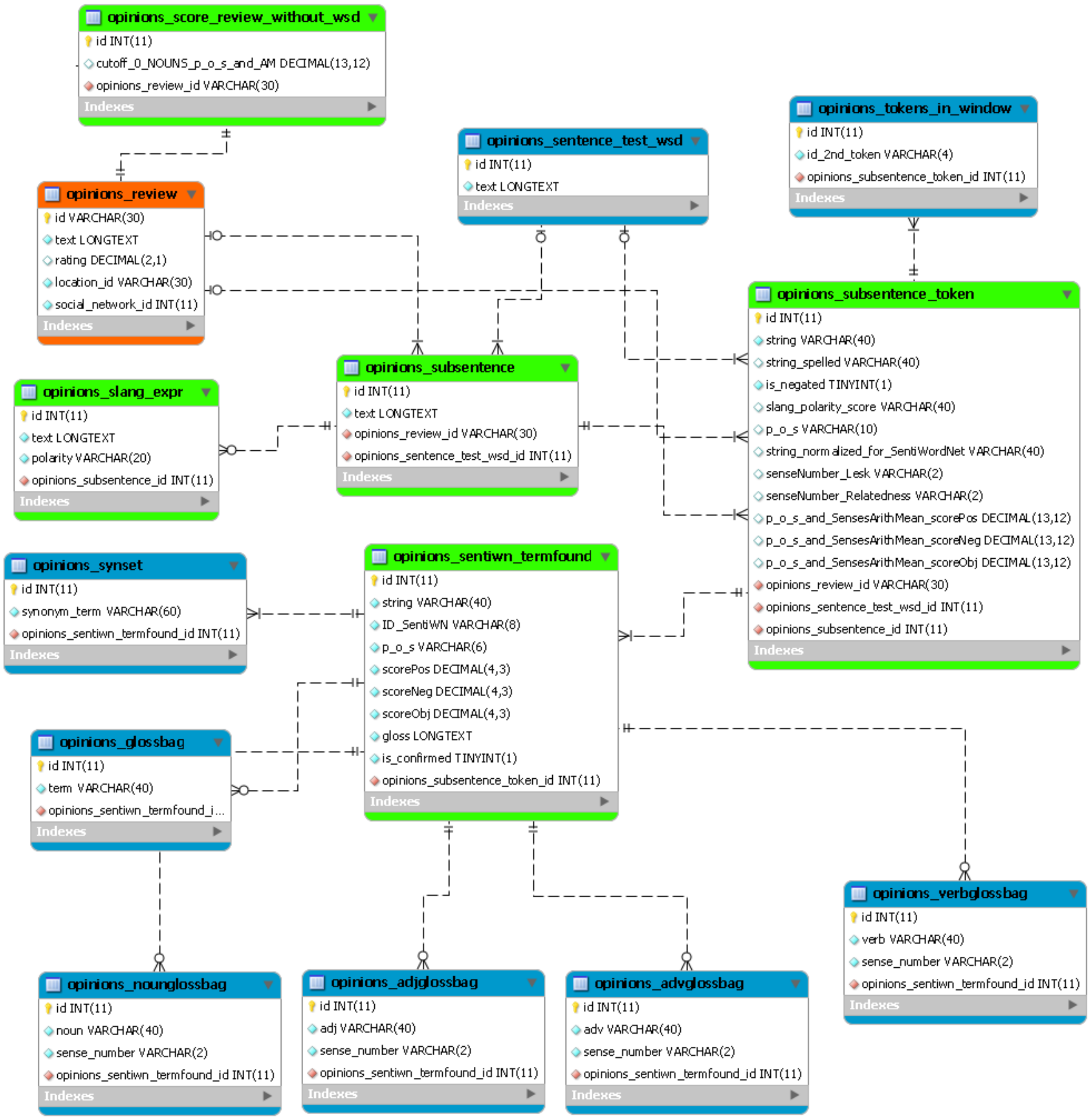

Figure 5.11: EER Diagram 4th Database's Version.

5.3.6 System Architecture

The *System Architecture* that has been described in Section 5.2.5 holds steady. The *Pipeline of the Sentiment Analysis* has been changed to include the refinement steps reported in Section 5.3.1. *Sentence splitting*, *tokenization* and *pos-tagging* are modified in order to improve the accuracy of the sentiment classification, and to permit the inclusion of the new features. *Exclamations, emoticons and slang expressions searching* are linked to tokenization and pos-tagging, while *negation detection* relates on sentence splitting. A function for *spelling correction* has also been developed, as depicted in Figure 5.12. Althought the Word Sense Disambiguation is currently not involved in the sentiment analysis, *eXtended WordNet* and the *WSD module* have been already setted up in the system for a further employment.

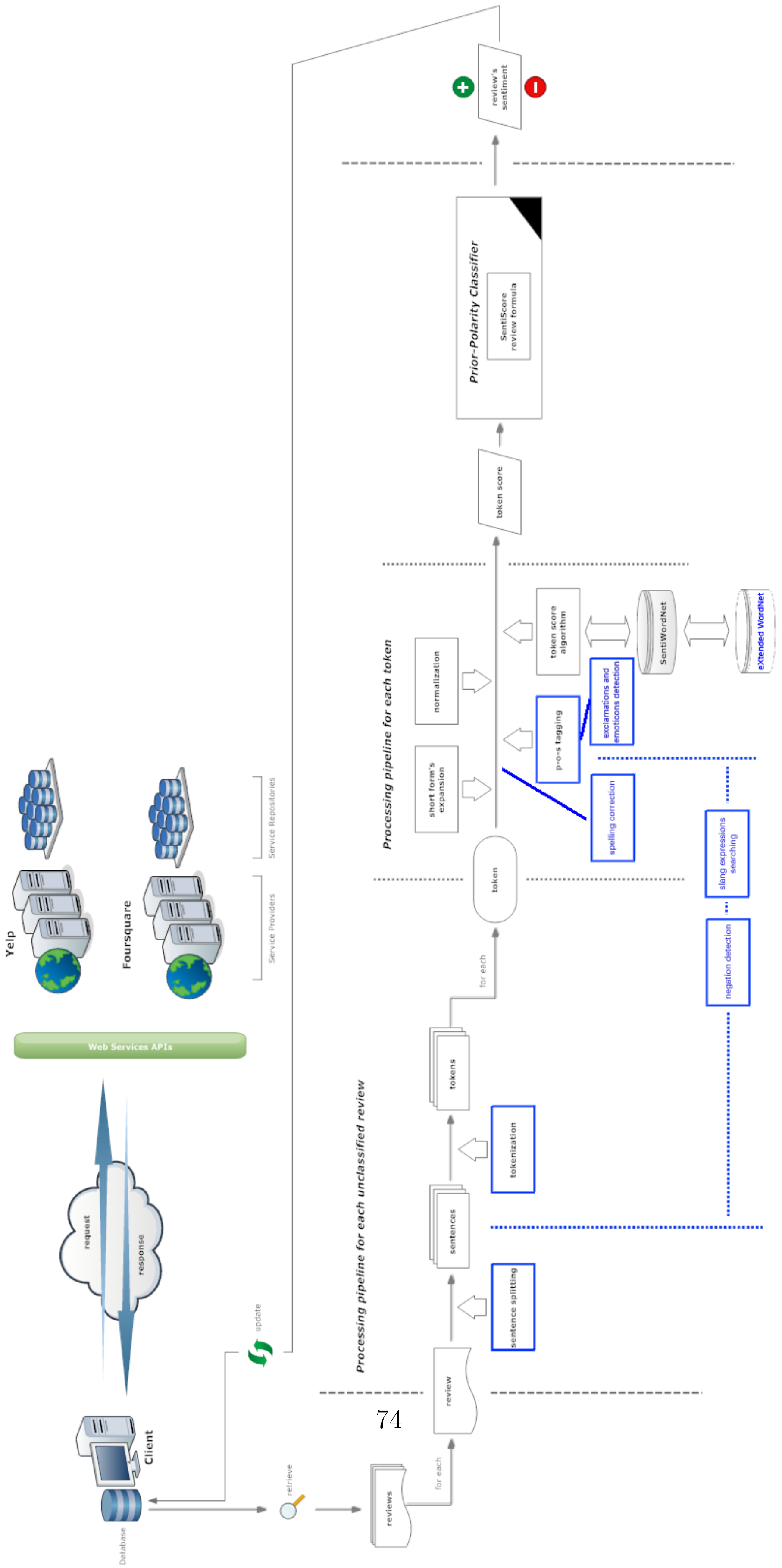

Figure 5.12: System Architecture and Sentiment Analysis Pipeline Refine-

Chapter 6

Implementation

6.1 Programming Language, Framework and Tools

6.1.1 Python, Django, NLTK

Python[1] is the programming language used until now in this project.

Python is a free object-oriented programming language that has many advantages to work with. It is simple to learn, it is portable in many platforms and can be integrated with many languages and programs, having a brief and clear syntax and offering many libraries. I am currently using the version 2.7 downloadable at `http://www.python.org/download/`

Moreover, for my prototype I exploited a framework: **Django**[2]. Django is an open source framework written in Python for the creation of web 2.0 applications, following the "three-tier" Model-View-Controller architectural pattern shown in Figure 6.1. The framework is useful in order to *control* the flow of the information; for example, calls to the social networks' web services (request), accessed through the APIs, were launched from some *views*; the obtained data (response) were saved into the database referring to the *model*. The three-tier architecture permits to have a modular software development and maintainance, to improve scalability and reusable of components. Django provides support for several DBMS, like MySql, Oracle, Postegre Sql,

[1] 1990-2010, Python Software Foundation

[2] 2005-2010 Django Software Foundation

Sqlite, and facilitates database's management providing an integrated ORM (Object-Relational Mapping) that avoid the manual writing of SQL queries for CRUD (Create Read Update Delete) operations.

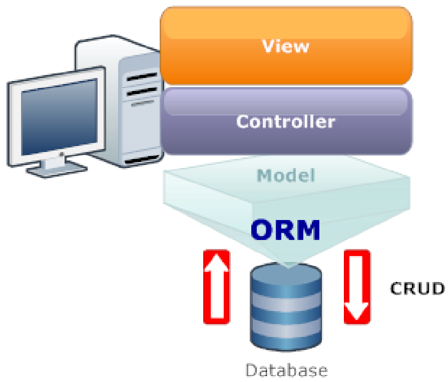

Figure 6.1: Django "three-tier" Architecture.

NLTK[3] (Natural Language ToolKit) is an open source library to deal with Natural Language Processing in Python; it offers already implemented Python modules to be able to execute many common operations, such as tokenization or part-of-speech tagging. It is distributed for Windows, Mac OSX and Linux. It includes a complete documentation, examples, demonstrations and data for experiments or training of classifiers.

Other examples of libraries for Natural Language Processing are GATE and LingPipe, but they were excluded because both implemented in Java.

6.1.2 APIs and Output Format (JSON)

API stands for Application Programming Interface. In general, remote systems grant access to their remote (web) services through APIs, giving the possibility to include their functionality into external applications or web sites. The *Social Networks* APIs of *Yelp* and *Foursquare* grant access to

[3]http://www.nltk.org/

their own services, giving the possibility to interact with them and to extract data. I used the APIs in order to principally retrieve information about reviews and corresponding locations.

Yelp and Foursquare APIs allow developers to receive the required data in either XML or JSON formats. I decided to manage the JSON output format. To integrate it with Python it is necessary to install an encoder/decoder package; the one that I utilized is the “simplejson 2.1.1”.

JSON (JavaScript Object Notation) is a data-interchange format based on the JavaScript Programming Language. It was developed using principles of C-family's languages, including C, C++, C#, Java, JavaScript, Perl, Python; for this reason, it results to programmers easy to read and easy to parse. It is language independent too. It works on array of values, and objects that are couples of name/value. A value can be a string in double quotes, or a number, or true or false or null, or an object or an array. The structures can be nested. For further information consult `http://www.json.org/`.

6.1.3 MySQL

MySQL[4], developed by the swedish company MySQL AB, stands for "My Structured Query Language" and is the most common open source relational database management system available on the net. It provides support for almost all the SQL syntax and it is compatible with many major programming languages including Python. In this project I use the version 5.0 of Mysql freely downloadable at `http://www.mysql.com/downloads/`.

To connect Python to a MySQL database, a MySQLdb library is required; in this project that is “MySQL-python-1.2.3” and it is available at `http://sourceforge.net/projects/mysql-python/`.

[4]2010, Oracle Corporation

Chapter 7

Experimentation

7.1 Evaluation of Sentiment Classification

7.1.1 Dataset

Experiments were conducted on a dataset of 400, 200 positive plus 200 negative, reviews, representing a subset of the data collected during the *opinion extraction* phase.

In the specific case, reviews rated with 5 or 4 stars and reviews rated with 1 or 2 stars were used as positive and negative respectively.

Reviews with rating 3 were excluded from the evaluation; because of being in the middle of the 5-star rating scale, they were treated as objectives. As it was suggested in Section 4.1, it has been taken advantage of the rating in order to avoid the hand-labeling of data needed for the evaluation purpose.

It is important to notify that rated reviews are just the ones extracted from Yelp; unfortunately, they are text excerpt, or better they represent text that usually is truncated and very short; this aspect has to be considered relevant in affecting the performance in effectiveness of the classifier.

7.1.2 Evaluation Metrics

The effectiveness of an IR system is commonly evaluated using two measures: Precision and Recall ((cf. Eq. (7.1)) and Eq. (7.2)); both of them compare the retrieved documents with the relevant documents.

(I) **Precision**:

It expresses the number of retrieved documents that are relevant over the number of retrieved documents, a percentage of how many documents are relevant among the retrieved ones.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{7.1}$$

(II) **Recall**:

It expresses the number of retrieved documents that are relevant over the number of relevant documents present in the data source, a percentage of relevant documents retrieved among the relevant ones.

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{7.2}$$

The previous notions can be made clear by examining the contingency Table 7.1.

Table 7.1: Relevance/Retrieval contingency table [MRS08].

	relevant	nonrelevant
retrieved	true positives (tp)	false positives (fp)
not retrieved	false negatives (fn)	true negatives (tn)

According to that, Precision and Recall can be redefined as:

(I) **Precision**:

$$P = tp/(tp + fp) \tag{7.3}$$

(II) **Recall**:

$$R = tp/(tp + fn) \tag{7.4}$$

These two parameters are necessary for establishing the effectiveness of classification. The information above ((cf. Eq. (7.3)) and Eq. (7.4)) can then be used to calculate the *accuracy* of the system, given by:

$$\textbf{Accuracy} = (tp + tn)/(tp + fp + fn + tn) \tag{7.5}$$

It is important to make clear that what is called "document" in the definitions of *Precision* and *Recall*, in this project is referred to a review; therefore, true positives and true negatives can be seen as the correct classifications considering positive and negative reviews respectively. Since the above metrics evaluate the capability of the classifier in identifying positive instances, *Precision* and *Recall* can then be splitted in *Positive Precision* ($Prec_p$ cf. Eq. (7.3)) and *Positive Recall* (Rec_p cf. Eq. (7.6)), *Negative Precision* ($Prec_n$ cf. Eq. (7.4)) and *Negative Recall* (Rec_n cf. Eq. (7.7)):

(I*) **Negative Precision**:

$$P = tn/(tn + fn) \tag{7.6}$$

(II*) **Negative Recall**:

$$R = fn/(tp + fn) \tag{7.7}$$

Mantaining the two measures separately it is important to see how the classifier behaves in the two different cases. Its *accuracy* can be reformulated in terms of number of correct classifications over number of all classifications.

$$\textbf{Accuracy} = \frac{N(correct\ classifications)}{N(all\ classifications)} \tag{7.8}$$

Precision and Recall can be combined in such a called *F-measure*, representing the harmonic mean of the two measurements, *"that assesses precision/recall tradeoff"* [OA]:

$$\textbf{F-score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7.9}$$

7.1.3 Computational Methods and Results

Computational Methods and Results refer to the *two* different *Sentiment Classifiers* which have been designed and evaluated in this work: the *Baseline* and its *Refinement*.

The Sentiment Classifiers have the goal to distinguish *Reviews* between *positive* and *negative* depending on their *Score*.

In order to be able to perform *Review Scores* it was necessary to extract *Tokens* and compute a *triple* of positive, negative and objecive *Scores* for each of them; about the 400 reviews considered, more than 10000 tokens were extracted.

Baseline Classifier

The pre-processing operations involved in the *Tokens'* extraction and the steps being part of the *Reviews' Score* calculation were already described in Section 5.2.1 and Section 5.2.3 respectively.

It has to be remembered from Section 5.2.3 that, to compute the triple of *Token Scores*, three different strategies were investigated : *Random Sense*, *All Senses Arithmetic Mean*, *P-O-S matching Senses Arithmetic Mean*; we will refer to them in terms of **Token Score's methods**. They can be seen as a superclass of the three polarity classes: positive, negative, objective; consequenlty, at the end of this computational step, for each token we will have nine different scores.

Token Scores will be used to compute the overall sentiment **Review's Score**, identified as *SentiScore(R)*, given by Equation 5.2. If *SentiScore(R)* is higher (lower, respectively) than zero, then R is labeled as positive (negative, respectively).

In order to discover the best procedure for classification, as it is described in Section 5.2.3, combinations of criteria such as *nouns' exclusion* and *cut-off*, were applied to Equation 5.2. We will refer to them in terms of **Review Score's methods**. Moreover, several cut-off points were tried. In [Mej10], for example, the best accuracy was reached with a 0.8 cut-off, although the cut-off approach was different from ours; they considered words that have a positive or negative polarity greater than the established cut-off; the size of the SentiWordNet lexicon at the cut-off point of 0.8 is reduced from 52,902 to 924. We judged this paper's approach too strict to be applied for our short reviews; therefore, we decided to apply two quite low cut-offs of 0.3 and 0.5 on a less restrictive approach: a token T, belonging to a review R, to be considered in the computation of its *SentiScore(R)*, has to pass the condition *"`ObjScore(T) < 1-(cutoff)`"*.

We can conclude saying that criteria were applied at both Token and Review level. At the end of first experiments, we obtained 18 different *Sentiment Scores* for each *Review*, corresponding to 18 *classifier models*, coming out from the combinations of the 3 *Token Score's methods* with 6 different *Review Score's methods.*

Table 7.2: Baseline Classifier's evaluation results: Precision

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
$Prec_n$	cut-off=0	48,5%	55%	56,5%
	cut-off=0, no-nouns=true	49%	53%	53,5%
	cut-off=0.3	52,5%	56%	57,5%
	cut-off=0.3, no-nouns=true	48,5%	52%	55%
	cut-off=0.5	52%	54%	50,5%
	cut-off=0.5, no-nouns=true	49%	50%	48,5%
$Prec_p$	cut-off=0	65,5%	68%	65%
	cut-off=0, no-nouns=true	61%	68%	65,5%
	cut-off=0.3	49,5%	55%	53%
	cut-off=0.3, no-nouns=true	44%	47%	52%
	cut-off=0.5	32%	32%	35%
	cut-off=0.5, no-nouns=true	29,5%	26,5%	35%

Table 7.3: Baseline Classifier's evaluation results: Recall

Metric	Review Score's Methods	Token Score's Methods		
		Random	all Senses AM	P-O-S, AM
Rec_n	cut-off=0	44%	39,82%	40,092%
	cut-off=0, no-nouns=true	45,54%	40,87%	41,518%
	cut-off=0.3	48,97%	44,44%	44,5%
	cut-off=0.3, no-nouns=true	53,93%	50,526%	46,4%
	cut-off=0.5	60%	41,02%	58,58%
	cut-off=0.5, no-nouns=true	63,35%	65,36%	59,5376%
Rec_p	cut-off=0	55,98%	60,18%	59,907%
	cut-off=0, no-nouns=true	54,464%	58,8745%	58,482%
	cut-off=0.3	51%	55,555%	55,497%
	cut-off=0.3, no-nouns=true	46,07%	49,47%	53,608%
	cut-off=0.5	40%	41,025%	41,42%
	cut-off=0.5, no-nouns=true	36,646%	34,64%	40,462%

Enhanced Classifier

Tables 7.2, 7.3, 7.4, 7.5, summarizes the Baseline Classifier performance measures in terms of *Precision*, *Recall*, *F-Measure*, overall *Accuracy* respectively.

The tables show that the more complex *Token Score Method* classifier performs better than the simpler one. Best results were obtained when combined **none cut-off** with a *Token Score Method* based on **P-O-S matching Senses**. Therefore, the Enhanced Classifier was setted up with those parameters.

The overall sentiment **Review's Score** is always computed using the *SentiScore* formula, described by Equation 5.2.

In this case, we experimented with an *unique classifier model* which is enriched with all the new features reported in Section 5.3.1.

7.2 Testing the Disambiguator

The experiments were carried out on some sentences retrieved from the SemCor corpus available in the NLTK package. *"SemCor is a 200,000 word*

Table 7.4: Baseline Classifier's evaluation results: F-Measure

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
$F - score_n$	cut-off=0	46,14%	46,195%	46,9%
	cut-off=0, no-nouns=true	47,2%	46,15%	46,75%
	cut-off=0.3	50,67%	49,555%	50,17%
	cut-off=0.3, no-nouns=true	51,071%	51,252%	50,335%
	cut-off=0.5	55,714%	44,9%	54,24%
	cut-off=0.5, no-nouns=true	55,26%	56,657%	53,455%
$F - score_p$	cut-off=0	60,37%	63,8514%	62,35%
	cut-off=0, no-nouns=true	57,547%	63,106%	61,8%
	cut-off=0.3	50,24%	55,276%	54,22%
	cut-off=0.3, no-nouns=true	45,011%	48,203%	52,79%
	cut-off=0.5	35,555%	35,955%	37,94%
	cut-off=0.5, no-nouns=true	32,7%	30,03%	37,53%
$F - score$	cut-off=0	53,255%	55,0232%	54,625%
	cut-off=0, no-nouns=true	52,373%	54,628%	54,275%
	cut-off=0.3	50,455%	52,4155%	52,195%
	cut-off=0.3, no-nouns=true	48,041%	49,7275%	51,5625%
	cut-off=0.5	45,6345%	40,4275%	46,09%
	cut-off=0.5, no-nouns=true	43,98%	43,3435%	45,4925%

sense–tagged sample of text, about 80% of which comes from the Brown Corpus" [PBP03].

Since the Word Sense Disambiguation computation required a huge amount of terms' comparisons and therefore of time, we decided to restrict the dataset to 30 test sentences.

In fact, considering a context window of n (i.e., equals to 9) words, the disambiguation algorithm needs to compare *(n-1)* pair of words to disambiguate a target word; if m are the words in a sentence/review than *((n-1)*m)* comparisons are necessary in order to disambiguate all the words; and if *((n-1)*m*s) senses* are considered * *5 gloss-bags* are constructed * *w words are in the gloss-bags*, *((n-1)*m*s*5*w)* terms' comparisons for a sentence need to be executed.

Table 7.5: Baseline Classifier's evaluation results: Accuracy

Metric	*Review Score's Methods*	*Token Score's Methods*		
		Random	all Senses AM	P-O-S, AM
Accuracy	cut-off=0	57%	61,5%	60,75%
	cut-off=0, no-nouns=true	55%	60,5%	59,5%
	cut-off=0.3	51%	55,5%	55,25%
	cut-off=0.3, no-nouns=true	46,25%	49,5%	53,5%
	cut-off=0.5	42%	43%	42,75%
	cut-off=0.5, no-nouns=true	39,25%	38,25%	41,75%

Table 7.6: Enhanced Classifier's evaluation results: Precision

Metric	*Review Score's Method*	*Token Score's Method: P-O-S, AM*
$Prec_n$	cut-off=0 + Enhanced Features	59,5%
$Prec_p$	cut-off=0 + Enhanced Features	61%

Precision and Recall of the Word Sense Disambiguation algorithm can be estimated according to the scoring policy of Senseval-2 [MSPG02].

(WSD) **Precision**:

$$P = correct/(wrong + correct) \tag{7.10}$$

(WSD) **Recall**:

Table 7.7: Enhanced Classifier's evaluation results: Recall

Metric	*Review Score's Method*	*Token Score's Method: P-O-S, AM*
Rec_n	cut-off=0 + Enhanced Features	40%
Rec_p	cut-off=0 + Enhanced Features	60%

Table 7.8: Enhanced Classifier's evaluation results: F-Measure

Metric	*Review Score's Method*	*Token Score's Method: P-O-S, AM*
$F - score_n$	cut-off=0 + Enhanced Features	48%
$F - score_p$	cut-off=0 + Enhanced Features	60,5%
$F - score$	cut-off=0 + Enhanced Features	54,5%

Table 7.9: Enhanced Classifier's evaluation results: Accuracy

Metric	*Review Score's Method*	*Token Score's Method: P-O-S, AM*
Accuracy	cut-off=0 + Enhanced Features	60%

$$R = correct/(wrong + correct + unattempted) \tag{7.11}$$

The Extended Lesk technique correctly identified the only 17% of terms' senses; 34% of the senses resulted wrong, while in the 49% of the cases the disambiguation failed. Therefore, it was achieved a Precision of 33%, a Recall of 17% and a F-Score equals to 22,5%.

The technique based on WordNet semantic Relatdness worked much better; it disambiguated in the right way the 40% of the terms, in a wrong way the 47% of the considered cases, and only 13% of terms' senses was unattempted. 46%, 40% and 42,8% were the values obtained corresponding to Precision, Recall and F-Measure respectively.

If we compare both the similarities, relatedness measures relate to more general concepts; they are less strict and they are more able to catch relationships also if the terms are slightly related; therefore it is possible to achieve better Recall results; but at the same time, also high probability of failure. The extended Lesk metric is potentially more accurate than previouses, since it searches for exact overlaps between senses; anyway, it needs to be applied in contexts longer than our short reviews in order to perform well; low Recall is then registered.

Mirroring the results stored in the database, we can furthermore say that, if we consider cases in which both the measures disambiguate terms' senses, the 44% of times they agree and the 56% they disagree; a further work could then consists in investigating a combined approach between them.

Moreover it was noticed that most of the failed cases relate on the attempt of disambiguating noun referring to first names or surnames.

Comparable results to ours can be, for example, the 32% of accuracy score reported by an Adapted version of the Lesk algorithm [BP02], that uses a context window of 3 words; other works relate to different settings; in [NB07] 85% of Precision was achieved in a disambiguation performed considering a context window of the size of three sentences, while in [MSPG02] 100 content words surrounding the target word were involved.

Chapter 8

Conclusions, Error Analysis and Further Work

Sentiment Analysis is a subfield of text analysis concerning with the extraction of emotional content in text. *"Polarity-annotated lexicons are one of the most frequently used resource in these studies"* [Mej10]. We have implemented a Prior-Polarity Classifier that exploit the SentiWordNet resource for a rule-based reviews' classification.

One of the limitations of this approach is its reliance on a fix classification rule based on considering prior polarities of only individual words; actually the algorithm is not able to adapt its behaviour to the *domain* and to *contextual polarity*. We are aware that the employed classification model is not sufficient for accurate sentiment classification. Since words can change their sentiment combined with others, a more correct rule-based approach should involve: the application of a generated set of patterns; the analysis of context and domain, where both can influence a word attitude; optionally, the complementary employ of set of consecutive words (n-grams of higher-order than employed unigrams, such as bigrams) to better capture patterns of sentiment expressions.

Our method achieved classification accuracy comparable to previous Sentiment Classification researches described in Section 3.2, althought they adopted different domains and approaches.

Tables 7.2, 7.3, 7.4, 7.5, summarizes the classifier performance measures in terms of *Precision*, *Recall*, *F-Measure*, overall *Accuracy* respectively.

Differences on reviews' classification performance between positive and negative, as measured by accuracy and other measures, can be attributed to a cause mentioned in [OT09], where it is described as reviewers can include negative remarks on positive opinions for a more balanced assessment, or viceversa they can choose to build up the expectation of a general good view to later postpone a negative impression. Afterwards the use of negative (or positive) terms in positive (or negative respectively) reviews can affect recall and other results.

From the experiment point of view, *low accuracies* might be due to the limited number of opinionated words contained in the reviews collected; reviews used for the evaluation purpose are text excerpt, or better they represent text that usually is truncated and very short; this aspect has to be considered relevant in affecting the performance in effectiveness of the classifier.

Sentiment Analysis is a challenging task, also for the *ironic words, colloquial language and expressions* that are used in writing reviews.

As explained before, most of the errors come from the wrong assignment of prior sentiment scores to the words, where words that have a polarity in SentiWordNet can have a different-opposite polarity in a considered review context. Other inaccuracies derive from the assignment of *part-of-speech* tags; for example, in a phrase such as "What a cool place", the term 'cool' is wrongly tagged as proper noun (NNP), and consequently identified by SentiWordNet as being objective, instead of positive (adjective). More imprecisions result from SentiWordNet scores, as researchers *found a few words which are always used to express positive feelings (e.g., love, enjoy, favorite, perfect, and great) or negative feelings (e.g., horrible, weak, useless, stupid, and silly), but their objective scores were assigned greater than 0.5 in SentiWordNet* [DZC10].

Mirroring the results of the *Baseline Classifier*, the more complex *Token Score Method* classifier performs better than the simpler one. Best results were obtained when combined none *cut-off* with a *Token Score Method* based on P-O-S matching Senses. Therefore, the *Enhanced Classifier* was setted up with those parameters.

Althought we *enriched the Classifier* with the new extended features described in Section 5.3.1, we did not registered any improvement in the performance of the system; the value of *F-Measure* and *Accuracy* remained *quite steady*. But then, if we look at the *Precision*, the difference between scores in negative and positive Precision results more slight; the algorithm seems to perform in a quite similar way for both the polarities; the employ of the *negation detection* may have helped to make the algorithm more stable. Currently, *emoticons and slang expressions* have not influence in determining polarities since they have been checked as appearing only few times (2 and 60 respectively) within the reviews' dataset.

Experimental results confirmed that Word Sense Disambiguation can improve sentiment classification performance; moreover, they indicated that all the words potentially carry emotions, including nouns.

A direction for further implementation would be to refine and extend the *rule-patterns for sentiment classification*, where the module designed for *Word Sense Disambiguation* could be included, however not before than another further study in depth about the issue.

At the moment, considering the *Word Sense Disambiguation procedure*, the 40% of Precision obtained from the technique based on our intuition of exploiting WordNet semantic similarities and eXtended WordNet is reasonable and exhaustive if compared to related WSD works. Moreover, we believe that, taking care of parameters, such as the *context window*, and analyzing *other relatedness measures*, results will be improved. That has been demonstrated by other researches like [PBP03] and [BH01]; we remark still one more time that Path-Based similarities measures have been chosen because of the shortest time in computation; however *other relatedness measure* can be easily incorporated in the system. Semantic *relations between synsets* (e.g., hypernymy, hyponymy, antonym, etc.) could be a source of information for the disambiguation. Since it has been noted that the two techniques usually differ in the choice of the correct sense, an *hybrid solution* may be another idea for a future WSD.

In future work the rule-based classifier may be *compared* with supervised learning techniques-approaches such as Support Vector Machines, Naive Bayes or Maximum Entropy. Other than a comparison of a rule-based and a machine-learning based approach to sentiment classification, these two approaches may be *combined* to improve sentiment-classification performance.

Moreover other lexical resources, such as Wordnet Domains could be employed to address the problem of the different menanings of words depending on the *domain* (e.g. the adjective "unpredictable" close to "movie" or "steering's behaviour").

The granularity in terms of sentiment reviews' scores may be capitalize in a future reviews' ranking process; where reviews may be ranked by how positive/negative they are. The ability of analyse sentiment of reviews retrieved from social networks, to classify and rank them, may be used to build a recommender application for interesting places.

A future improvement may consist in establishing a feature extraction procedure, for the detection of target entities and, at the same time, of the sentiments referred to them; this more specific task will allow to understand what is liked or disliked about entities (i.e. interesting places and their features such as food or atmosphere).

Finally, a further work may be to evaluate a "classification range" similarly to [MSPN10] where an empirical study was conducted with the intent of establishing *"a sentiment range for determining whether a review R should be treated as positive, negative, or neutral"*.

Bibliography

[art04] Word sense disambiguation of wordnet glosses. *Computer Speech & Language*, 18(3):301 – 317, 2004. Word Sense Disambiguation.

[ASPPBL06] Henry Anaya-Sánchez, Aurora Pons-Porrata, and Rafael Berlanga-Llavori. Word sense disambiguation based on word sense clustering. In Jaime Sichman, Helder Coelho, and Solange Rezende, editors, *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*, volume 4140 of *Lecture Notes in Computer Science*, pages 472–481. Springer Berlin / Heidelberg, 2006.

[BES10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[BH01] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the*

Association for Computational Linguistics, Pittsburgh, USA, 2001.

[BP02] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 136–145, London, UK, UK, 2002. Springer-Verlag.

[BP03] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 805–810, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[CBH+01] Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301 – 310, 2001.

[CCD+07] S. Cerini, V. Compagnoni, A. Demontis, M. Formentelli, and G. Gandini. *Language resources and linguistic theory: Typology, second language acquisition, English linguistics (Forthcoming)*, chapter Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. Franco Angeli Editore, Milano, IT, 2007.

[CZ05] Pimwadee Chaovalit and Lina Zhou. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches. *Hawaii International Conference on System Sciences*, 4:112c, 2005.

[DBL08] *Proceedings of the 24th International Conference on Data Engineering Workshops, ICDE 2008, April 7-12, 2008, Cancún, México*. IEEE Computer Society, 2008.

[DC01] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *In Asia Pacific Finance Association Annual Conf. (APFA)*, 2001.

[Den08] Kerstin Denecke. Using sentiwordnet for multilingual sentiment analysis. In *ICDE Workshops* [DBL08], pages 507–512.

[Den09] Kerstin Denecke. Are sentiwordnet scores suited for multi-domain sentiment classification? In Grosky et al. [GAP09], pages 33–38.

[DZC10] Yan Dang, Yulei Zhang, and HsinChun Chen. A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews. *IEEE Intelligent Systems*, 25:46–53, 2010.

[EG04] Jonas Ekedahl and Koraljka Golub. Word sense disambiguation using wordnet and the lesk algorithm, 2004. Class Report. Language Processing and Computational Linguistics.

[ES06] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, pages 417–422, 2006.

[Esu06] Andrea Esuli. Opinion Mining. Language and Intelligence Reading Group, June 14, 2006. Presentation slides, retrieved September 30, 2010, from `http://citeseerx.ist.psu.edu`.

[Esu08a] Andrea Esuli. *Automatic Generation of Lexical Resources for Opinion Mining: Models, Algorithms and Applications.* PhD in Information Engineering, PhD School "Leonardo da Vinci", University of Pisa, 2008. Retrieved November 18, 2010, from `http://etd.adm.unipi.it/theses/available/etd-04042008-163914/unrestricted/andrea_esuli_phdthesis.pdf`.

[Esu08b] Andrea Esuli. Automatic generation of lexical resources for opinion mining: models, algorithms and applications. *SIGIR Forum*, 42:105–106, November 2008.

[FK08] Angela Fahrni and Manfred Klenner. Old Wine or Warm Beer: Target-Specific Sentiment Analysis of Adjectives. In *Proc.of the Symposium on Affective Language in Human and Machine, AISB 2008 Convention, 1st-2nd April 2008. University of Aberdeen, Aberdeen, Scotland*, pages 60 – 63, 2008.

[GAP09] Bill Grosky, Frédéric Andrès, and Pit Pichappan, editors. *Fourth IEEE International Conference on Digital Information Management, ICDIM 2009, November 1-4, 2009, University of Michigan, Ann Arbor, Michigan, USA*. IEEE, 2009.

[GJM10] Steven Grijzenhout, Valentin Jijkoun, and Maarten Marx. Opinion mining in dutch hansards. In *Proceedings Workshop From Text to Political Positions (t2pp 2010)*, April 2010.

[GSNZ06] A Goryachev, M Sordo, L Ngo, and QT Zeng. Implementation and evaluation of four different methods of negation detection. Technical report, DSG, 2006.

[HM97] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

[JM07] Daniel Jurafsky and James H. Martin. *Speech and Language Processing, An introduction to natural language processing, computational linguistics, and speech recognition.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007.

[KB08] Sopan Govind Kolte and Sunil G. Bhirud. Word sense disambiguation using wordnet domains. In *Proceedings of the 2008*

First International Conference on Emerging Trends in Engineering and Technology, pages 1187–1191, Washington, DC, USA, 2008. IEEE Computer Society.

[KB09] S. G. Kolte and S. G. Bhirud. Exploiting links in wordnet hierarchy for word sense disambiguation of nouns. In *Proceedings of the International Conference on Advances in Computing, Communication and Control*, ICAC3 '09, pages 20–25, New York, NY, USA, 2009. ACM.

[KFP09] Manfred Klenner, Angela Fahrni, and Stefanos Petrakis. PolArt: A Robust Tool for Sentiment Analysis. 2009.

[KI06] Alistair Kennedy and Diana Inkpen. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2):110–125, 2006.

[Les86] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM.

[Liu] Bing Liu. Opinion Mining and Summarization (including review spam detection), tutorial given at WWW-2008, April 21, 2008 in Beijing, China. Retrieved September 30, 2010, from `http://www.cs.uic.edu/~liub/FBS/opinion-mining-sentiment-analysis.pdf`.

[Liu10a] Bing Liu. Sentiment analysis: A multifaceted problem. *IEEE Intelligent Systems, 25(3)*, pages 76–80, 2010.

[Liu10b] Bing Liu. Sentiment analysis and subjectivity. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010. ISBN 978-1420085921.

[LSM95] Xiaobin Li, Stan Szpakowicz, and Stan Matwin. A wordnet-based algorithm for word sense disambiguation. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, pages 1368–1374, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[MBMP10] Tamara Martin, Alexandra Balahur, Andrés Montoyo, and Aurora Pons. Word Sense Disambiguation in Opinion Mining: Pros and Cons. *Special issue: Natural Language Processing and its Applications. Journal on Research in Computing Science*, 46:119–130, 2010.

[Mej10] Yelena Mejova. Tapping into sociological lexicons for sentiment polarity classification. *4th Russian Summer School in Information Retrieval*, pages 14–27, September 2010.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[MSPG02] Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. The role of domain information in word sense disambiguation. *Nat. Lang. Eng.*, 8:359–373, December 2002.

[MSPN10] Rani Qumsiyeh Maria Soledad Pera and Yiu-Kai Ng. An unsupervised sentiment classifier on summarized or full reviews. In *In Proceedings of the 11th International Conference on Web Information Systems Engineering (WISE 2010)*, pages 142–156, Hong Kong, December 2010. Springer.

[NB07] Sudip Kumar Naskar and Sivaji Bandyopadhyay. Word sense disambiguation using extended wordnet. In *Proceedings of the International Conference on Computing: Theory and Applications*, pages 446–450, Washington, DC, USA, 2007. IEEE Computer Society.

[NKW05] Jc Na, C Khoo, and Phj Wu. Use of negation phrases in automatic sentiment classification of product reviews. *LIBRARY COLLECTIONS ACQUISITIONS TECHNICAL SERVICES*, 29(2):180–191, 2005.

[NSI08] Nicolas Nicolov, Franco Salvetti, and Steliana Ivanova. Sentiment Analysis: Does Coreference Matter? *Symposium on Affective Language in Human and Machine*, 2008.

[NSK+04] Jin-Cheon Na, Haiyang Sui, Christopher Khoo, Syin Chan, and Yunyun Zhou. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Conference of the International Society for Knowledge Organization (ISKO)*, pages 49–54, 2004.

[NY03] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: capturing favorability using natural language processing. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77, New York, NY, USA, 2003. ACM.

[OA] Daniel Ortiz-Arroyo. Lecture notes: Information retrieval. part ii: The vector space model, metrics in ir. Retrieved April 21, 2010, from `http://www.cs.aaue.dk/~do/Teaching/S10/IR/notes2.pdf`.

[OK09] Tim O'Keefe and Irena Koprinska. Feature selection and weighting methods in sentiment analysis. In *Proceedings of the Fourteenth Australasian Document Computing Symposium*, New South Wales, Sydney, december 2009. School of Information Technologies, University of Sydney.

[OT09] B. Ohana and B. Tierney. Sentiment classification of reviews using SentiWordNet. In *9th. IT & T Conference*, page 13, 2009.

[PBP] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2588 of *Lecture Notes in Computer Science*, pages 241–257. Springer Berlin / Heidelberg.

[PBP03] Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan. Maximizing semantic relatedness to perform word sense disambiguation, 2003.

[PGS+10] Georgios Paltoglou, Stéphane Gobron, Marcin Skowron, Mike Thelwall, and Daniel Thalmann. Sentiment analysis of informal textual communication in cyberspace. In *proceedings of ENGAGE 2010*, pages 13–25, 2010.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, 2008.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[Pot] Chris Potts. Sentiment lexicons. Linguist 287 / CS 424P: Extracting Social Meaning and Sentiment, Fall 2010, Sep 28. Handout on sentiment lexicons, retrieved October 4, 2010, from `http://www.stanford.edu/class/cs424p/materials/ling287-handout-09-28-classification.pdf`.

[PT09] Rudy Prabowo and Mike Thelwall. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143 – 157, 2009.

[PZ06] Livia Polanyi and Annie Zaenen. Contextual valence shifters. In W. Bruce Croft, James Shanahan, Yan Qu, and Janyce Wiebe, editors, *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10. Springer Netherlands, 2006.

[Res95] Philip Resnik. Disambiguating noun groupings with respect to wordnet senses. In *Proceedings of the third workshop on very large corpora*, pages 54–68, 1995.

[RPB04] Ganesh Ramakrishnan, B. Prithviraj, and Pushpak Bhattacharya. A gloss-centered algorithm for disambiguation. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 217–221, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[RygBP05] Paolo Rosso, Manuel Montes y gómez, Davide Buscaldi, and Luis Villaseñor Pineda. Two web-based approaches for noun sense disambiguation. In *In: Int. Conf. on Comput. Linguistics and Intelligent Text Processing, CICLing-2005, Springer Verlag, LNCS (3406), Mexico D.F*, pages 261–273, 2005.

[San96] Mark Sanderson. *Word sense disambiguation and information retrieval.* PhD Thesis, Technical Report (TR- 1997-7) of the Department of Computing Science, University of Glasgow G12 8QQ, UK, 1996.

[SDSO66] Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis.* MIT Press, 1966.

[TBP+10] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61:2544–2558, December 2010.

[TNKS09] Tun Thura Thet, Jin-Cheon Na, Christopher S.G. Khoo, and Subbaraj Shakthikumar. Sentiment analysis of movie reviews on discussion boards using a linguistic approach. In *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, TSA '09, pages 81–84, New York, NY, USA, 2009. ACM.

[Tur02] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[WWH05] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[WWH09] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics*, 0(0):1–35, 2009.