UNIVERSITÀ DI PISA
Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
Ingegneria dell'Informazione

Tesi di Dottorato di Ricerca

Automatic Generation of Lexical Resources for Opinion Mining: Models, Algorithms and Applications

Andrea Esuli

Anno 2008

UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"

Corso di Dottorato di Ricerca in
Ingegneria dell'Informazione

Tesi di Dottorato di Ricerca

Automatic Generation of Lexical Resources for Opinion Mining: Models, Algorithms and Applications

Autore:

Andrea Esuli

Relatori:

Dott. Fabrizio Sebastiani

Prof. Luca Simoncini

Anno 2008

Abstract

Opinion mining is a recent discipline at the crossroads of Information Retrieval and of Computational Linguistics which is concerned not with the topic a document is about, but with the opinion it expresses. It has a rich set of applications, ranging from tracking users' opinions about products or about political candidates as expressed in online forums, to customer relationship management. Functional to the extraction of opinions from text is the determination of the relevant entities of the language that are used to express opinions, and their opinion-related properties. For example, determining that the term `beautiful` casts a *positive* connotation to its subject.

In this thesis we investigate on the automatic recognition of opinion-related properties of terms. This results into building opinion-related *lexical resources*, which can be used into opinion mining applications. We start from the (relatively) simple problem of determining the *orientation* of subjective terms. We propose an original semi-supervised term classification model that is based on the quantitative analysis of the *glosses* of such terms, i.e. the definitions that these terms are given in on-line dictionaries. This method outperforms all known methods when tested on the recognized standard benchmarks for this task.

We show how our method is capable to produce good results on more complex tasks, such as discriminating *subjective* terms (e.g., `good`) from *objective* ones (e.g., `green`), or classifying terms on a fine-grained *attitude* taxonomy.

We then propose a relevant refinement of the task, i.e., distinguishing the opinion-related properties of distinct *term senses.* We present SentiWordNet, a novel high-quality, high-coverage lexical resource, where each one of the 115,424 senses contained in WordNet has been automatically evaluated on the three dimensions of positivity, negativity, and objectivity.

We propose also an original and effective use of random-walk models to rank term senses by their positivity or negativity. The random-walk algorithms we present have a great application potential also outside the opinion mining area, for example in word sense disambiguation tasks. A result of this experience is the generation of an improved version of SentiWordNet.

We finally evaluate and compare the various versions of SentiWordNet we present here with other opinion-related lexical resources well-known in literature, experimenting their use in an *Opinion Extraction* application. We show that the use of SentiWordNet produces a significant improvement with respect to the baseline system, not using any specialized lexical resource, and also with respect to the use of other opinion-related lexical resources.

Acknowledgments

These three years have been a very formative experience for me, both professionally and personally.

My first big thank goes to Fabrizio Sebastiani. He has been not only a great advisor, which has shared with me his wide experience, but also a challenging partner to work with on the many engaging research problems we have faced in these years. Thanks to Luca Simoncini, for his discreet-but-sure presence over these years.

My gratitude goes to all the persons I have collaborated with during these three years. Tiziano Fagni, a perfect teammate in the developement of many, many lines of code. Ilaria Clara Urciuoli, for her invaluable annotation work that has produced the I-CAB Opinion corpus. Michał Pryczek, for our discussions on the evaluation measures for information extraction. Shlomo Argamon and Kenneth Bloom, for the pleasant collaboration in the work on determining attitude type of terms.

Finally I thank my family. My father and my mother, for having grown me the way I am. My wife Cristina, for everithing else.

A Caterina e Cristina.

Contents

Introduction

"Public opinion is the thermometer a monarch should constantly consult."

Napoleone Bonaparte

I.1 It's all about opinions

Opinions have a relevant impact on our everyday life. Opinions give us information about how reality is perceived by other people. We use opinions to express our points of view.

We are used to ask and compare opinions from other people to support our decisions. For example, we usually read some movie reviews to decide which movie to rent. We are interested in other people's opinion about us. We like to contribute to discussions, giving advices expressing our opinions. For example, participating in online forums or newsgroups.

Large organizations and industries are also interested in opinions. For example, political parties constantly track the trend of public opinion by means of polls. Industries perform market surveys collecting people's opinions about their products and the ones of their competitors in order to compare them and plan their market strategies.

There are many sources from which opinions can be gathered such as newspapers, television, and the Internet. Internet is probably the most valuable one, given its widespread diffusion, accessibility and liberality. Online forums, newsgroups, blogs, and specialized sites provide millions of information feeds from which opinions can be extracted. Obviously such a large amount of input data cannot be handled by a single person, or even a group, without the use of automatic processing tools that allow to filter and discriminate the relevant information from the irrelevant one.

Starting about eight years ago, this practical need for opinion analysis tools has met the interest of researchers working in the fields of Information Retrieval and/or Computation Linguistics and a new discipline, Opinion Mining, has gradually emerged.

I.2 Opinion Mining

The Opinion Mining (OM) discipline places itself at the crossroads of Information Retrieval and of Computational Linguistics, two disciplines from which OM gathers, and combines, many concepts, ideas and methods. OM is concerned with the analysis of the opinions expressed in documents. OM is thus a *non-topical* text analysis discipline, i.e. a discipline which is not concerned with the topic of the analyzed document but with some other of its properties, which in the case of OM are the opinions expressed about its subject matter. Other non-topical text analysis disciplines are, for example, *authorship attribution* (i.e. recognizing the author of a document in a set of candidates) and *genre recognition* (i.e. recognizing the type of a document, for example an editorial, a scientific paper, an informal letter).

The name “Opinion Mining” [23, 25, 33, 42, 68] is not the only one used in literature, many other names have been used such as “Opinion Analysis” [37, 80], “Sentiment Classification” [39, 77], “Sentiment Analysis” [71, 75], and “Affect Analysis” [87]. All these names can be considered as near-synonyms to generally identify the discipline, each one possibly denoting some specific subtasks. For example, the name Sentiment Classification is commonly used in works focusing on the task of classifying entire documents as having a positive or a negative connotation. We choose to use the OM name because it currently identifies the latest direction of the discipline, which is the identification and analysis of the properties of each single opinion expression within a document.

I.2.1 OM tasks

Regardless of the name used to identify a specific task, it is possible to classify OM tasks along two dimensions: the opinion-related *dimensions* analyzed and the *granularity* of the analysis.

Dimensions analyzed in OM tasks

The opinion-related dimensions most frequently investigated in the literature are *subjectivity* and *orientation.*

In a subjectivity analysis task the goal is to determine which of the analyzed content contains subjective expressions [75, 101] (i.e. expressions that are not open to an objective interpretation [73]) or has a factual nature instead. For example, the sentence “`The iPhone has a great interface`” contains a subjective evaluation, while the sentence “`The iPhone weights 135 grams`” reports an objective fact.

Given some subjective content, the problem of determining its orientation consists in recognizing if it expresses a positive or a negative evaluation with respect to its subject matter [77, 91]. For example, the sentence “`My laptop has a powerful graphic card`” contains a positive evaluation, while the sentence “`This mobile phone lacks many features`” contains a negative evaluation.

Another dimension which has been frequently investigated is *force*, i.e. the evaluation of the intensity of the various opinion expressions [76, 98, 99]. The analysis of this dimension is usually associated with one of the previous two. For example, one could compare the relative intensity of the two sentences “`X is a good tool for blogging`” and “`Y is a perfect tool for blogging`” to conclude that the second one has a *higher* force with respect to the positivity they express.

Other dimensions of opinion expressions, which have been less investigated, are typically related to a finer classification of opinion expressions [4, 88, 94]. For example, the distinction between the *aesthetic* evaluation expressed by the term `elegant`, and the *moral* evaluation expressed by the term `generous`.

Granularity of OM tasks

The analysis of the previously described opinion-related dimensions can be performed at various levels of granularity.

Some applications consider a whole *document* as a single entity, and determine some global opinion-related properties. For example, discriminating the documents containing opinion expressions from those not containing them [101]. Other applications instead analyze such properties for each distinct *sentence*, considering that a document could contain a mix of facts and evaluations, possibly with different orientation [75].

Going in even more detail, single *propositions*, or arbitrarily defined *text spans*, can be identified in text as having some opinion-related properties, such as expressing a subjective concept, or being the holder of a subjective expression [97].

The most detailed levels of analysis are concerned with the analysis of the opinion-related properties of each distinct *term* [44, 53, 92], or even *term sense* [3, 32, 48]. Such analysis is typically performed considering the term out of the context of any of its specific use in any document, with the goal of determining its general opinion-related properties (ORPs). For example, determining that the term `success` has a positive orientation, the term `disaster` has a negative orientation, and the term `triangular` instead does express an objective concept.

The analysis performed at the level of detail of documents, sentences, or smaller text portions is typically driven by a specific applicative goal (e.g. to perform a structured extraction of the opinions expressed in the given documents). The analysis at the term and term sense level of detail has instead the more general goal to understand which elements of the language express subjectivity, and how they are used. A typical result of such analysis is the generation of *OM lexical resources* (e.g. a list of terms which have a given orientation). Such lexical resources are used by the OM applications as sources of information to better identify subjectivity in text, thus improving their effectiveness.

I.3 OM applications

OM literature has been largely driven by applicative interest in domains such as mining online corpora for opinions, or customer relationship management.

What do people think about the latest camera-equipped cellular phone? What is the general opinion on the just-passed governmental decree on safety on the workplace? Is popular support for presidential candidate X's promise of a tax cut growing? Systems capable of automatically detecting and tracking the evolution of customers' opinions concerning a given product, of voters' thoughts on a political candidate, of citizens' opinions on governmental policies, are thus of enormous help to marketers, social scientists, information analysts, policy enforcers, and opinion makers, since they enable them to examine (and draw statistical information from) quantities of textual data beyond the reach of manual approaches.

One of the earlier OM works is the one from Das and Chen [22], where the global trend of orientation of opinions toward a certain company, posted in online forums, is compared to its stock price in the same time interval, showing a relevant correlation between the two measures.

Turney [91] has worked on classifying the orientation of product reviews as either "thumbs up" or "thumbs down", by using a measure of semantic association between the content of documents and two small sets of terms which are deemed to be representative of the two categories under examination.

Similarly, Pang et al. [77] have investigated the use of a standard supervised text classification approach to classify movie reviews by their orientation. On the same task, Pang and Lee [75] have used a subjectivity classifier of sentences to build documents summaries which contain only the opinionated content. They have used it as a preprocessing component to the movie review classifier, improving its effectiveness. Pang and Lee [76] have also proposed an extension of the movie review classification task into an *ordinal regression* task, i.e. the assignment of movie reviews to a rating scale, ranging from no star (totally negative review) to five stars (totally positive review).

On the movie review classification task, Whitelaw et al. [94] have shown a significant accuracy improvement produced by using a lexical resource where subjective terms are classified by their *attitude type* (e.g. distinction between aesthetic, affective, and moral evaluations).

Attardi and Simi [5] have used information about which terms express subjectivity in a text search engine. Their system is able, for example, to handle the query `[‘‘Barack Obama’’ near` *`Subjective`*`]` which retrieves all the document where the expression “`Barack Obama`” appears near (within a maximum distance of a given number of words) any subjective term. They have used such system in the 2006 TREC Blog track [74] showing a relevant improvement in the precision of retrieval of opinionated content derived from the use of such kind of queries.

Yu and Hatzivassiloglou [101] describe a system which classifies entire documents, and then each of their sentences, as subjective or not. Its intended use is as the first data processing block in an *opinion question answering* system. Such system is designed to answer questions like “What are the causes of global warming?” where the answer has to take into account the multiple perspectives, and opinions, on the topic. This is a different, and harder, task with respect to traditional question answering in which answers are typically factual and univocal.

Wiebe et al. [95, 97] have investigated the *opinion extraction* problem, i.e. the task of detecting, *within* a sentence or document, the exact expressions denoting the statement of an opinion, and detecting therein the sub-expressions denoting the key components and properties (e.g. the opinion holder, the object of the opinion, the type of opinion, the strength of the opinion, etc.) within this statement. From these initial works, many other researchers have then worked on the task [11, 18, 17, 55, 56].

The works mentioned here, which are just a very small sampling of the many currently published on the OM topic [26], show the wide range of possible applications related to opinions, and the various approaches to them.

I.4 Lexical resources for OM

A common point in almost any work on OM is the need to identify which elements of language contribute to express the subjectivity in text. Such identification if often accomplished by using a lexical resource that lists the relevant ORPs of lexical items. The lexical items in a lexical resource can be both single words or multiword sequences [94].

Unfortunately, nowadays there are just a few of such resources [44, 92, 94], they have a very small coverage of the language (about 3,600 terms for the largest one), and they are all related to the English language. This is mostly due the fact that their manual compilation has a great cost. The result of this situation is that, besides the research on OM applications, the other main line of OM research is focused on the automatic generation of such lexical resources.

This thesis is devoted to the investigation of methods for the automatic generation of lexical resources for OM. Our focus will be not on the final OM applications, at least not directly, but on the construction of the resources that such applications can use to improve their results.

Our investigation will go into two directions: (a) the generation of lexical resources that are *correct* with respect to the ORPs under examination, (b) the generation of resources that have a high *coverage* of the language (i.e. analyzing the largest number of elements of the language for their ORPs).

I.5 Structure of the thesis

This thesis is composed by six main chapters. In each chapter we focus on a specific problem related to OM lexical resources (except the last where we actually evaluate their use in a practical application). All the chapters have a similar structure: introduction and definition of the task, review of related works, description of the proposed solution, definition of the experimental setup, experiments, discussion of results, and conclusions.

In the first part of this thesis we focus on methods for automatically determining the ORPs of *terms*.

We start in Chapter 1 with the (relatively) simple task of determining the *orientation* of a set of given *subjective* terms, e.g. labeling as Positive terms like `good`, `happy`, `beautiful`, which cast a positive connotation onto the expression they qualify, and labeling as Negative terms like `bad`, `sad`, `ugly`, which instead cast a negative connotation onto the expression they qualify.

The original method we propose is based on:

- the hypothesis that the terms with similar orientation tend to have similar *glosses* in a dictionary;
- the application of text classification methods to glosses, in order to classify the terms the glosses refer to.

We experimentally compare our method with the others published in the literature. The results presented in this chapter have been originally published in [30].

In Chapter 2 we move to the more difficult task of recognizing also the *subjectivity* of terms, following the observation that an automatic method able to classify terms by orientation is of little value if it requires to already know which terms are subjective.

We thus present an extension of the method that classifies terms by orientation to recognize also subjectivity. This method can be used to automatically build subjectivity (and orientation) lexicons from scratch, which is of great applicative interest. The results presented in this chapter have been originally published in [31].

In Chapter 3 we apply our term classification method on even more detailed sentiment-related taxonomies, which label, for example, the term `beautiful` as an aesthetic evaluation and the term `honest` as a moral one. The results presented in this chapter have been originally published in [4].

In the second part of the thesis we focus on methods for automatically determining the ORPs of *term senses*.

Distinguishing among the various senses of a term allows one to build more accurate resources, relaxing the two alternative assumptions usually made in term-based lexical resources that the properties assigned to a term (a) are related to its most frequent meaning, or (b) indicate that at least one of its possible meanings has such properties. For example, the term `blue` is typically used to mean the color, but can also refer to the negative mood. Similarly, the term `estimable` may be used to describe a measurable quantity, or to describe a respectable person.

In Chapter 4 we leverage on our last experience on the determination of the ORPs of terms, and apply our term classification method to determine the subjectivity and orientation of term senses. As a result of this activity we produce SentiWordNet, a lexical resource which assigns a positivity and a negativity score to each of the more than 115,000 term senses defined in WordNet, a rich lexical database of the English language. The results presented in this chapter extend those originally published in [32].

In Chapter 5 we propose another original method for determining the ORPs of term senses. This new method is based on using random-walk models on the link graph determined by the occurrences of term senses in the glosses defining other term senses. As the random-walk models we use PageRank, the famous ranking algorithm used by the Google search engine to rank Web pages, and two variants of it based on different definitions of the link relation, and thus different random-walk models. The results presented in this chapter extend those originally published in [33, 34]. Combining the results obtained in this chapter and the ones obtained in the previous one we present also a new version of SentiWordNet, which improves over the previous one.

In Chapter 6 we test the impact of using the lexical resources presented in the previous chapters in an *Opinion Extraction* (OE) task. The OE task consists in recognizing the exact part of text in document where any expression of opinion or emotion appears, along with the opinion holder. We will use in our experiments two corpora of news where opinion expressions have been annotated using a special markup language, and our task will be to automatically reproduce such annotations. The two corpora mainly differ for the language of their documents, English for one corpus and Italian for the other. This allow us to show an effective cross-language use of the SentiWordNet resource. The results presented in this chapter are still unpublished.

In Chapter 7 we conclude, reporting a summary of our results and depicting the possible future path of the research in OM.

Chapter 1

Determining the Semantic Orientation of Terms through Gloss Classification

Abstract

In this chapter we present a novel method for determining the orientation of subjective terms. The proposed method is based on the quantitative analysis of the *glosses* of such terms, i.e. the definitions that these terms are given in on-line dictionaries, and on the use of the resulting term representations for semi-supervised term classification. The presented method outperforms all known methods when tested on the recognized standard benchmarks for this task.

1.1 Introduction

When executing an Opinion Mining task, lexical resources have a relevant role in the identification of opinion expressions and the evaluation of their properties. Such resources are typically composed by *sets of terms*, usually two: one of Positive terms which identify those terms that give a positive connotation to the text where they appear in (e.g. `good, nice, beautiful`), and one of Negative terms which identify those terms that give a negative connotation instead (e.g. `bad, ugly,` `disgusting`). Terms that do not appear in any of the two sets are implicitly considered to be Objective, and to not contribute to the overall orientation of text[1].

Some resources may also identify the *strength* of the positivity (or negativity) of each Positive (resp. Negative) term. For example, this can be done by defining a function $O(t)$ that, given a term t, returns a value in $[-1, 1]$, where the sign identifies the positivity or negativity of the term and the absolute value the strength of its orientation [53, 54, 92]. Other lexical resources may instead subdivide terms in a finite number of sets, each one identifying a specific strength of orientation (e.g. Strong-Positive, Weak-Positive...), but it is easy, and also a common practice, to derive a discrete-valued $O(t)$ function from such resources [2].

Lexical resources have a relevant role in the development of systems that perform opinion-related tasks on text. The early work from Hatzivassiloglou and Wiebe [45] shows that just checking the presence of adjectives known to have relevant opinion-related properties (i.e. positive or negative orientation) is a strong clue to the identification of sentences that express some subjectivity, while just checking for the presence of adjectives produces a much lower performance.

Turney's [91] method to classify documents as either Positive or Negative is one of the simplest, yet effective, examples of use of an OM lexical resource. It is based on considering the algebraic sum of the orientation scores of terms as representative of the orientation of the document they belong to:

$$\text{Classify}(d) = \begin{cases} \text{Positive} & \sum_{t \in T(d)} O(t) > 0 \\ \text{Negative} & \sum_{t \in T(d)} O(t) < 0 \\ \text{Neutral} & \sum_{t \in T(d)} O(t) = 0 \end{cases} \tag{1.1}$$

where d is the document to be classified and $T(d)$ returns the terms contained in the document.

More sophisticate approaches are also possible. For example, in *text classification* tasks based on machine learning methods, a typical use

[1] Actually, due to the typical lack of *coverage* of the lexical resources, some of the terms not appearing in them may have instead a positive or a negative orientation. In this chapter we do not address the coverage issue, which will be faced in the following chapters.

Table 1.1: Bethard et al. [9] results.

Features	Precision	Recall
BoW	50.97%	43.17%
BoW + HL	50.00%	43.72%
BoW + AL	**54.27%**	**48.63%**

of lexical resources is to generate new *features*, to be added to those commonly used to represent a document (i.e. words appearing in the document itself). Each of the new features has the role to identify a specific property of the document, e.g. "the document contains at least 5 Positive terms", thus (implicitly) modeling the intuition "if the document contains many positive terms, then it is probably a Positive document". Then it will be up to the machine learning algorithm, during the learning process, to evaluate if the new features are relevant or not for the classifier being built.

The work of Bethard et al. [9] is an example of an OM system which uses some lexical resource-based features to perform automatic annotation of opinions in text. The goal of their work is to classify propositions as expressing opinions or not. They have used a typical machine learning approach, based on Support Vector Machines (SVM). The vectorial representation of documents to be fed to the learner, for training, and then to the classifier, for testing, are built using various features, which also include two sets of opinion related features: a hand-built list of *opinion-oriented* 2,973 terms (HL, in Table 1.1), and an automatically generated version composed by 24,929 terms (AL, in Table 1.1). In the experimental activity the performance of the proposition classifier in various configurations is compared, including one with a simple *bag-of-word* (BoW, in Table 1.1) model, and two which also use the above mentioned lexical resources. Results (see Table 1.1) show a relevant improvement from the use of the automatically generated lexical resource. In the Discussion section of [9], the authors' comment is:

> **. . . our classification was significantly improved by using lists of opinion words which were automatically derived with a variety of statistical methods, and these extended lists proved more useful than smaller, more accurate manually constructed lists.**

`delicious` : **greatly pleasing** or **entertaining**

`nice` : **pleasant** or **pleasing** or **agreeable** in nature or appearance

`disturbing` : causing **distress** or **worry** or **anxiety**

`bogus` : **fraudulent**; having a **misleading** appearance

Figure 1.1: Excerpt from WordNet glosses of the terms `delicious`, `nice`, `disturbing` and `bogus`.

Building lexical resources by hand is a very time-consuming task and, until now, it has only been done to compile relatively small resources (e.g. one of the largest is the lexicon of the General Inquirer with 3,596 terms, see Section 1.4.1), which are typically used as *benchmark* for evaluation of automatic methods. In fact, OM literature has concentrated on developing automatic methods to build such resources (the most relevant of them are described in detail in Section 1.2).

In this chapter we present our first contribution to the topic of automatically generating lexical resources for OM, by proposing a novel method for determining the orientation of terms. The method relies on the application of semi-supervised learning to the task of classifying terms as belonging to either the Positive category or the Negative category. The novelty of the method lies in the fact that it exploits a source of information which previous techniques for solving this task had never attempted to use, namely, the *glosses* (i.e. textual definitions) that the terms have in an online "glossary", or dictionary. Our basic assumption is that terms with similar orientation tend to have "similar" glosses: for instance, that the glosses of `delicious` and `nice` will both contain appreciative expressions, while the glosses of `disturbing` and `bogus` will both contain derogative expressions (see Figure 1.1).

The method is semi-supervised (see e.g. [72]), in the sense that

1. a small training set of "seed" Positive and Negative terms is chosen for training a term classifier;

2. before learning begins, the training set is enriched by navigating through a thesaurus, adding to the Positive training terms (i) the terms related to them through relations (such as e.g. synonymy) indicating similar orientation, and (ii) the terms related to the Negative training terms through relations (such as e.g. antonymy) indicating opposite orientation (the Negative training terms are enriched through an analogous process).

We test the effectiveness of our method on the three benchmarks previously used in this literature, and first proposed in [44, 53, 92], respectively. The proposed method is found to outperform the previously known best-performing method [92] in terms of accuracy, although by a small margin. This result is significant, notwithstanding this small margin, since our method is computationally much lighter than the previous top-performing method, which required a space- and time-consuming phase of Web mining.

1.1.1 Chapter outline

In Section 1.2 we review in some detail the related literature on determining the orientation of terms. The methods and results presented in this section are analyzed and taken as reference in Section 1.3, which describes our own approach to determining the orientation of terms, and in Section 1.4 and 1.5, which report on the experiments we have run and on the results we have obtained. Section 1.6 concludes.

1.2 Related work

1.2.1 Hatzivassiloglou and McKeown

The work of Hatzivassiloglou and McKeown [44] has been the first to deal with the problem of determining the orientation of terms. The method attempts to predict the orientation of (subjective) *adjectives* by analyzing pairs of adjectives (conjoined by `and`, `or`, `but`, `either-or`, or `neither-nor`) extracted from a large unlabeled document set. The underlying intuition is that the act of conjoining adjectives is subject to linguistic constraints on the orientation of the adjectives involved (e.g. `and` usually conjoins two adjectives of the same orientation, while `but` conjoins two adjectives of opposite orientation). This is shown in the following three sentences (where the first two are perceived as correct and the third is perceived as incorrect) taken from [44]:

1. `The tax proposal was simple and well received by the public.`

2. `The tax proposal was simplistic but well received by the public.`

3. (*) `The tax proposal was simplistic and well received by the public.`

Their method to infer the orientation of adjectives from the analysis of their conjunctions uses a three-step supervised learning algorithm:

1. All conjunctions of adjectives are extracted from a set of documents.

2. The set of the extracted conjunctions is split into a training set and a test set by placing in the test set only the conjunctions such that each of the conjoined adjectives appears at least α times in conjunction with any other adjective occurring in the test set. The α parameter thus defines the hardness of the test set, since conjunctions of adjectives with fewer links have less associated information, and are thus harder to classify (see Table 1.2).

3. The conjunctions in the training set are used to train a classifier, based on a log-linear regression model, which classifies pairs of adjectives either as having the same or as having different orientation. The classifier is applied to the test set, thus producing a graph with the hypothesized same- or different-orientation links between all pairs of adjectives that are conjoined in the test set.

4. A clustering algorithm uses the graph produced in Step 3 to partition the test adjectives into two clusters.

5. By using the intuition that positive adjectives tend to be used more frequently than negative ones, the cluster containing the terms of higher average frequency in the document set is deemed to contain the Positive terms.

Table 1.2: Hatzivassiloglou and McKeown's results [44].

α	Number of adjectives in test set	Percentage of full test set	Accuracy
2	730	100.0%	78.08%
3	516	70.7%	82.56%
4	369	50.5%	87.26%
5	236	32.3%	92.37%

For their experiments, the authors used a term set consisting of 657/679 adjectives labeled as being Positive/Negative (hereafter, *the HM term set*). The document collection from which they extracted the conjunctions of adjectives is the unlabeled 1987 Wall Street Journal document set[2]. In the experiments reported in [44], and summarized in Table 1.2, the above algorithm determines the orientation of adjectives with an accuracy of 78.08% on the full HM term set. accuracies ranging from 78.08% to 92.37%, depending on the hardness of the test data as specified by the α parameter.

1.2.2 Turney and Littman

Turney and Littman [92] have approached the problem of determining the orientation of terms by bootstrapping from a pair of two minimal sets of "seed" terms:

S_p = {`good, nice, excellent, positive, fortunate, correct, superior`}

S_n = {`bad, nasty, poor, negative, unfortunate, wrong, inferior`}

which they have taken as descriptive of the two categories Positive and Negative.

Their method is based on computing the *pointwise mutual information* (PMI)

[2] Available from the ACL Data Collection Initiative as CD-ROM 1 (`http://www.ldc.upenn.edu/Catalog/`).

$$PMI(t, t_i) = \log \frac{\Pr(t, t_i)}{\Pr(t) \Pr(t_i)} \tag{1.2}$$

of the target term t with each seed term t_i as a measure of their semantic association. Given a term t, its orientation value $O(t)$ is given by

$$O(t) = \sum_{t_i \in S_p} PMI(t, t_i) - \sum_{t_i \in S_n} PMI(t, t_i) \tag{1.3}$$

i.e. by the sum of the weights of its semantic association with the seed positive terms minus the sum of the weights of its semantic association with the seed negative terms.

The authors have tested their method on the HM term set from [44] and also on the categories Positive and Negative defined in the General Inquirer lexicon [86]. The General Inquirer is a text analysis system that uses, in order to carry out its tasks, a large number of categories[3], each one denoting the presence of a specific trait in a given term. The two main categories are Positive/Negative, which contain 1,915/2,291 terms having a positive/negative orientation. Examples of positive terms are `advantage`, `fidelity` and `worthy`, while examples of negative terms are `badly`, `cancer`, `stagnant`. In their experiments the list of terms is reduced to 1,614/1,982 entries (hereafter, *the TL term set*) after removing terms appearing in both categories (17 terms – e.g. `deal`) and reducing all the multiple entries of a term in a category, caused by multiple senses, to a single entry.

Pointwise mutual information is computed using two methods, one based on IR techniques (PMI-IR) and one based on latent semantic analysis (PMI-LSA). In the PMI-IR method, term frequencies and co-occurrence frequencies are measured by querying a document set by means of a search engine with a "t" query, a "t_i" query, and a "t NEAR t_i" query, and using the number of matching documents returned by the search engine as estimates of the probabilities needed for the computation of PMI in Equation 1.2. In the AltaVista search engine[4], which was used in the experiments, the NEAR operator produces a match for a document when its operands appear in the document at a maximum distance of ten terms, in either order. This is a stronger constraint than

[3] The definitions of all such categories are available at `http://www.webuse.umd.edu:9090/`

[4] `http://www.altavista.com/`

the one enforced by the AND operator, that simply requires its operands to appear anywhere in the document.

In the experiments, three document sets were used for this purpose: (i) *AV-Eng*, consisting of all the documents in the English language indexed by AltaVista at the time of the experiment; this amounted to 350 million pages, for a total of about 100 billion term occurrences; (ii) *AV-CA*, consisting of the AV-Eng documents from `.ca` domains; this amounted to 7 million pages, for a total of about 2 billion term occurrences; and (iii) *TASA*, consisting of documents collected by Touchstone Applied Science Associates[5] for developing "The Educator's Word Frequency Guide" [102]; this amounted to 61,000 documents, for a total of about 10 million word occurrences.

The results of [92] (illustrated in Table 1.3) show that performance tends to increase with the size of the document set used; this is quite intuitive, since the reliability of the co-occurrence data increases with the number of documents on which co-occurrence is computed. On the HM term set, the PMI-IR method using AV-Eng outperformed by an 11% margin (87.13% vs. 78.08%) the method of [44]. It should be noted that, in order to avoid overloading the AltaVista server, only a query every five seconds was issued, thus requiring about 70 hours for downloading the AV-Eng document set. On the much smaller TASA document set PMI-IR was computed locally by simulating the behaviour of AltaVista's NEAR operator; this document set brought about a 20% *decrease* in accuracy (61.83% vs. 78.08%) with respect to the method of [44].

Table 1.3 also reports the results Using AND instead of NEAR on AV-Eng brought about a 19% decrease in accuracy compared to the use of NEAR on the TL term set (67.0% vs. 82.84%). The PMI-LSA measure was applied only on the smallest among the three document sets (TASA), due to its heavy computational requirements. The technique showed some improvement over PMI-IR on the same document set (a 6% improvement on the TL term set, a 9% improvement on the HM term set).

1.2.3 Kamps et al.

Kamps et al. [53] focused on the use of lexical relations defined in WordNet[6]. They defined a graph on the adjectives contained in the inter-

[5] `http://www.tasa.com/`

[6] `http://wordnet.princeton.edu/`

Table 1.3: Turney and Littman's results [92].

Term	Method	Operator	Doc set	Accuracy
set			Doc set	(%)
TL	PMI-IR	NEAR	AV-Eng	**82.84**
TL	PMI-IR	AND	AV-Eng	67.00
TL	PMI-IR	NEAR	AV-CA	76.06
TL	PMI-IR	NEAR	TASA	61.26
TL	PMI-LSA	NEAR	TASA	65.27
HM	PMI-IR	NEAR	AV-Eng	**87.13**
HM	PMI-IR	NEAR	AV-CA	80.31
HM	PMI-IR	NEAR	TASA	61.83
HM	PMI-LSA	NEAR	TASA	67.66

section between the TL term set and WordNet, adding a link between two adjectives whenever WordNet indicates the presence of a synonymy relation between them. On this graph, the authors defined a so-called "geodesic" distance measure $d(t_1, t_2)$ between terms t_1 and t_2, which amounts to the length of the shortest path that connects t_1 and t_2 (with $d(t_1, t_2) = +\infty$ if t_1 and t_2 are not connected). The orientation of a term is then determined by its relative distance from the two seed terms good and bad, i.e.

$$SO(t) = \frac{d(t, \texttt{bad}) - d(t, \texttt{good})}{d(\texttt{good}, \texttt{bad})} \tag{1.4}$$

The adjective t is deemed to belong to Positive iff $SO(t) > 0$, and the absolute value of $SO(t)$ determines, as usual, the strength of this orientation (the constant denominator $d(\texttt{good}, \texttt{bad})$ is a normalization factor that constrains all values of SO to belong to the $[-1, 1]$ range).

With this method, only adjectives connected to any of the two chosen seed terms by some path in the synonymy relation graph can be evaluated. This is the reason why the authors limit their experiment to the 663 adjectives of the TL term set (18.43% of the total 3,596 terms) reachable from either good or bad through the WordNet synonymy relation (hereafter, *the KA set).* They obtain a 67.32% accuracy value, which is not terribly significant given the small test set and the limitations inherent in the method.

1.2.4 Takamura et al.

Takamura et al. [89] determine term orientation (for Japanese) according to a "spin model", i.e. a physical model of a set of electrons each endowed with one between two possible spin directions, and where electrons propagate their spin direction to neighboring electrons until the system reaches a stable configuration. The authors equate terms with electrons and term orientation to spin direction. They build a neighborhood matrix connecting each pair of terms if one appears in the gloss of the other, and iteratively apply the spin model on the matrix until a "minimum energy" configuration is reached. The orientation assigned to a term then corresponds to the spin direction assigned to electrons.

1.2.5 Kim and Hovy

The system of Kim and Hovy [54] tackles orientation detection by attributing, to each term, a positivity score *and* a negativity score; interestingly, terms may thus be deemed to have both a positive and a negative correlation, maybe with different degrees, and some terms may be deemed to carry a stronger positive (or negative) orientation than others. Their system starts from a set of positive and negative seed terms, and expands the positive (resp. negative) seed set by adding to it the synonyms of positive (resp. negative) seed terms and the antonyms of negative (resp. positive) seed terms. The system classifies then a target term t into either Positive or Negative by means of two alternative learning-free methods based on the probabilities that synonyms of t also appear in the respective expanded seed sets. A problem with this method is that it can classify only terms that share some synonyms with the expanded seed sets.

The methods of [54, 89] are difficult to compare with our method and the other ones since they were not evaluated on publicly available datasets.

1.3 Determining the orientation of a term by gloss classification

We present a method for determining the orientation of a term based on the classification of its glosses.

Our use of glosses for reasoning about the semantics of terms is inspired by Michael Lesk's seminal word sense disambiguation method [61]. In this method, in order to determine which among the senses $\{s_1(t), \ldots, s_n(t)\}$ of a given term t a given occurrence $o(t)$ of t in a text belongs to, the n glosses $\{g(s_1(t)), \ldots, g(s_n(t))\}$ are ranked in terms of their similarity with the text window in which $o(t)$ belongs, and the top-ranked gloss is chosen as indicating the correct sense. For doing so, both the text window and the glosses are given vectorial representations through standard text indexing methods.

Our process for determining the orientation of terms is composed of the following steps:

1. Two seed sets Tr_p^0 and Tr_n^0, of terms that are representative of the two categories Positive and Negative, are provided as input.

2. Lexical relations (e.g. synonymy) from a thesaurus, or online dictionary, are used in order to find new terms that will also be considered representative of the two categories because of their relation with the terms contained in Tr_p^0 and Tr_n^0. This process can be iterated K times, using as input at every iteration k the result of the previous one $k-1$. The new terms, once added to the original ones, yield two new, richer sets Tr_p^K and Tr_n^K of terms; together they form the training set for the learning phase of Step 4.

3. For each term t_i in $Tr^K = Tr_p^K \cup Tr_n^K$ or in the test set (i.e. the set of terms to be classified), a textual representation of t_i is generated by collating all the glosses of t_i as found in a machine-readable dictionary[7]. Each such representation is converted into vectorial form by standard text indexing techniques.

4. A binary text classifier is trained on the terms in Tr^K and then applied to the terms in the test set.

In Step 1 it is possible to choose two relatively small sets of terms, as done in [92], or even two singleton sets, as done in [53].

Step 2, in which new representative terms are added to either Tr_p^k or Tr_n^k, is based on the hypothesis that the lexical relations used in this expansion phase, in addition to defining a relation of meaning, between

[7] In general a term t_i may have more than one gloss, since it may have more than one sense; dictionaries normally associate one gloss to each sense.

the two involved terms, also define a relation of orientation: for instance, it seems plausible that two synonyms may have the same orientation, and that two antonyms may have opposite orientation. This step is thus reminiscent of the use of the synonymy relation as made by Kamps et al. [53], and Kim and Hovy [54]. Any relation between terms that expresses, implicitly or explicitly, similar (e.g. synonymy) or opposite (e.g. antonymy) orientation, can be used in this process. It is possible to combine more relations together so as to increase the expansion rate (i.e. computing the union of all the expansions obtainable from the individual relations), or to implement a finer selection (i.e. computing the intersection of the individual expansions).

In Step 3, all the glosses associated to a term in a dictionary are collated so as to form a textual representation for the term. The the basic assumption is that terms with a similar orientation tend to have "similar" glosses: for instance, that the glosses of `delicious` and `nice` will contain both appreciative expressions, while the glosses of `disturbing` and `bogus` will contain both derogative expressions (see Figure 1.1). Note that, quite inevitably, the resulting textual representations will also contain "noise", in the form of the glosses related to word senses different from the ones intended[8].

Finally, Step 4 uses text classification techniques in order to obtain a model for the orientation of terms from the glosses of training terms, and then applies it to the classification of test terms.

Altogether, the learning method we use is *semi-supervised* (rather than supervised), since some (most) of the "training" data used have been labeled by our algorithm, rather than by human experts.

Performing gloss classification as a device for classifying the terms described by the glosses, thus combining the use of lexical resources and text classification techniques, has two main goals:

- taking advantage of the richness and precision of human-defined linguistic characterizations as available in lexical resources such as WordNet;

- enabling the classification of *any* term, provided there is a gloss for it in the lexical resource.

[8] Experiments in which some unintended senses and their glosses are filtered out by means of part-of-speech analysis are described in Section 1.5.

This latter point is relevant, since it means that our method can classify basically any term. This is in sharp contrast with e.g. the method of [44], which can only be applied to adjectives, and with that of [53], which can only be applied to terms directly or indirectly connected to the terms `good` or `bad` through the WordNet synonymy relation.

1.4 Experiments

1.4.1 Test sets and seed sets

We have run our experiments on the HM, TL, and KA term sets, which have been described in Sections 1.2.1, 1.2.2, and 1.2.3, respectively.

Respect to the HM corpus we have used the *full* set, corresponding to the value $\alpha = 2$ in Table 1.2.

The intersection between TL and HM consists of 652 terms; 641 of these have the same label on both corpora (a 98.31% agreement).

1.4.2 Seed terms

As discussed in Section 1.3, the method requires bootstrapping from two seed sets Tr_p^0 and Tr_n^0 representative of the categories Positive and Negative. In the experiments we have alternatively used the same seven positive and seven negative terms used in [92] (the *Tur* training set), as listed in Section 1.2.2, or the singleton sets {`good`} and {`bad`} (the *Kam* training set), as used in [53]. Note that *Kam* is a proper subset of *Tur*.

1.4.3 Expansion method for seed sets

We have used WordNet version 2.0 [40] as the source of lexical relations, mainly because of its ease of use for automatic processing. However, any thesaurus could be used in this process.

From the many lexical relations defined in WordNet, we have chosen to explore *synonymy* (*Syn*; e.g. `use / utilize`), *direct antonymy* (Ant^D; e.g. `light / dark`), *indirect antonymy* (Ant^I; e.g. `wet / parched`)[9],

[9] Indirect antonymy is defined in WordNet as antonymy extended to those pairs whose opposition of meaning is mediated by a third term; e.g. `wet / parched`, are indirect antonyms, since their antonymy relation is mediated by the similarity of `parched` and `dry`. It should be remarked that $Ant^D \subseteq Ant^I$.

hypernymy ($Hyper$; e.g. `car / vehicle`) and *hyponymy* ($Hypon$, the inverse of hypernymy; e.g. `vehicle / car`), since they looked to us the most obvious candidate transmitters of orientation. We have made the assumption that Syn, $Hyper$, and $Hypon$ relate terms with the same orientation, while Ant^D and Ant^I relate terms with opposite orientation.

The function *ExpandSimple*, which we have iteratively used for expanding (Tr_p^0, Tr_n^0), is described in Figure 1.2. The input parameters are the initial seed sets Tr_p^0 and Tr_n^0 to be expanded, the graph defined on all the terms by the lexical relation used for expansion, and a flag indicating if the relation expresses similar or opposite orientation between two terms related through it. The training set is built by initializing it to the seed sets (Step 1), and then by recursively adding to it all terms directly connected to training terms in the graph of the considered relation (Step 2)[10]. The role of Steps 3 and 4 is to avoid that the same term be added to both Tr_p^k and Tr_n^k; this is accomplished by applying the two rules:

Priority : if a term already belongs to Tr_p^{k-1} (resp. Tr_n^{k-1}), it cannot be added to Tr_n^k (resp. Tr_p^k);

Tie-break : if a term is added at the same time to both Tr_p^k and Tr_n^k, it is not useful, and can thus be eliminated from both.

The relations we have comparatively tested in seed set expansion are:

$Syn(J)$	synonymy, restricted to adjectives
$Syn(*)$	synonymy, regardless of POS
$Ant^D(J)$	direct antonymy, restricted to adjectives
$Ant^D(*)$	direct antonymy, regardless of POS
$Ant^I(J)$	indirect antonymy, restricted to adjectives
$Ant^I(*)$	indirect antonymy, regardless of POS
$Hypon(*)$	hyponymy, regardless of POS
$Hyper(*)$	hypernymy, regardless of POS

The function $ExpandMultiple$, described in Figure 1.3, combines several lexical relations into a single expansion step, by computing the union or the intersection of the expansions obtained according to the individual relations.

```
function ExpandSimple
Input :
    $Tr_p^{k-1}, Tr_n^{k-1}$ : input sets for the Positive and Negative categories
    $G_{rel}$ : graph defined on terms by the lexical relation $rel$
    $S_{rel}$ : boolean flag specifying if the relation expresses similarity
        or opposition of orientation
Output :
    $Tr_p^k, Tr_n^k$ : expanded seed sets
Body :
    1. $Tr_p^k \leftarrow Tr_p^{k-1}$; $Tr_n^k \leftarrow Tr_n^{k-1}$;
    2. foreach $term$ in $Tr_p^{k-1}$ do
         $Temp \leftarrow$ set of all terms directly connected to $term$ in $G_{rel}$;
         if $S_{rel}$ then
             $Tr_p^k \leftarrow Tr_p^k \cup Temp$;
         else
             $Tr_n^k \leftarrow Tr_n^k \cup Temp$;
       foreach $term$ in $Tr_n^{k-1}$ do
         $Temp \leftarrow$ set of all terms directly connected to $term$ in $G_{rel}$;
         if $S_{rel}$ then
             $Tr_n^k \leftarrow Tr_n^k \cup Temp$;
         else
             $Tr_p^k \leftarrow Tr_p^k \cup Temp$;
    3. $Tr_p^k \leftarrow Tr_p^k - Tr_n^{k-1}$; $Tr_n^k \leftarrow Tr_n^k - Tr_p^{k-1}$;
    4. $Dup \leftarrow Tr_p^k \cap Tr_n^k$; $Tr_p^k \leftarrow Tr_p^k - Dup$; $Tr_n^k \leftarrow Tr_n^k - Dup$;
```

Figure 1.2: Basic expansion function for seed sets.

```
function $ExpandMultiple$
Input :
    $Tr_p^{k-1}$, $Tr_n^{k-1}$ : input sets for the Positive and Negative categories
    $G = \{(G_{rel_1}, S_{rel_1}), \ldots, (G_{rel_n}, S_{rel_n})\}$ : list of graphs defined by
        the lexical relations $rel_1$, ..., $rel_n$
    $union$ : boolean flag specifying if union or intersection of relations
        has to be used in the expansion
Output :
    $Tr_p^k$, $Tr_n^k$ : expanded seed sets
Body :
    1. foreach $(G_{rel_i}, S_{rel_i})$ in $G$ do
         $(TempPos_i, TempNeg_i) \leftarrow$
             $ExpandSimple(Tr_p^{k-1}, Tr_n^{k-1}, G_{rel_i}, S_{rel_i})$;
    2. if $union$ then
         $Tr_p^k \leftarrow \bigcup_{i=1\ldots n} TempPos_i$;
         $Tr_n^k \leftarrow \bigcup_{i=1\ldots n} TempNeg_i$;
       else
         $Tr_p^k \leftarrow \bigcap_{i=1\ldots n} TempPos_i$;
         $Tr_n^k \leftarrow \bigcap_{i=1\ldots n} TempNeg_i$;
    3. $Dup \leftarrow Tr_p^k \cap Tr_n^k$; $Tr_p^k \leftarrow Tr_p^k - Dup$; $Tr_n^k \leftarrow Tr_n^k - Dup$;
```

Figure 1.3: Combination of basic expansions for seed sets.

Restricting a relation R to a given part of speech (POS) (e.g. adjectives) means that, among the terms related through R with the target term t, only those that have the same POS as t are included in the expansion. This is possible since WordNet relations are defined on word senses (*synsets*), rather than words, and since WordNet synsets are POS-tagged[11].

After evaluating the effectiveness of individual relations (see Section 1.5), we have chosen to further investigate the combination of the best-performing ones, i.e.:

$Syn(J) \cup Ant^D(J)$	union of $Syn(J)$ and $Ant^D(J)$
$Syn(J) \cap Ant^D(J)$	intersection of $Syn(J)$ and $Ant^D(J)$
$Syn(J) \cup Ant^I(J)$	union of $Syn(J)$ and $Ant^I(J)$
$Syn(J) \cap Ant^I(J)$	intersection of $Syn(J)$ and $Ant^I(J)$

and the corresponding $*$ versions (e.g. $Syn(*) \cup Ant^D(*)$), not restricted to adjectives.

In the experiments, we have used these relations iteratively, starting from the seed sets Tr_p^0 and Tr_n^0 and producing various chains of expansion, iterating until no other terms can be added to $Tr_p^k \cup Tr_n^k$[12].

1.4.4 Representing terms

The creation of textual representations of terms is based on the use of glosses extracted from a dictionary.

We have first experimented with the (freely accessible) online version of the Merriam-Webster dictionary[13] (*MW*). We have gathered the *MW* glosses by using a Perl script that, for each term, queries the *MW* site for the dictionary definition of the term, retrieves the HTML output from the server, isolates the glosses from the other parts of the document (e.g. side menus, header banner), and removes HTML tags. After this processing, some text unrelated to the glosses is still present in

[10] For non-symmetric relations, like hypernymy, the edge direction must be outgoing from the seed term.

[11] In the experiments reported in Section 1.4 the only restriction we test is to adjectives, since all the terms contained either in the *Tur* or in the *Kam* seed sets are adjectives.

[12] We have reached a maximum of $K = 16$ iterations for the Ant^D relation when used on the *Kam* seed set.

[13] `http://www.m-w.com/`

the resulting text, but more precise text cleaning would require manual processing, because of the extremely variable structure of the entries in MW. For this reason we have switched to WordNet, leaving the use of MW only to a final experiment on an optimized setting.

Glosses in WordNet have instead a regular format, that allows the production of cleaner textual representations. In WordNet, each concept is represented by a *synset*, which gathers all the terms that could represent the concept. Thus, a term belongs to many synset for many senses it has, and each different sense is marked by a unique number. For example, the adjective `unfortunate` has 3 senses and the second sense is represented by the synset {`unfortunate#J#2`,`inauspicious#J#1`} (Figure 1.4 shows the original WordNet entry for the term `unfortunate`). Synsets are partitioned on the four POS: adjectives, adverbs, nouns, verbs. Each synset has always a gloss associated to it. A gloss is composed by the list of the terms belonging to the synset (T), the concept definition (D) and, optionally, some sample phrases (S).

The textual representation for a term can be easily built by collating all the WordNet glosses for all the synsets it belongs to. We have tested four different methods for creating textual representations of terms. The first one puts together the synset terms and the definition (we dub it *the TD method*), while the second also includes the sample phrases (*the TDS method*); if the lexical relation used for expansion is limited to a given POS (e.g. adjectives), we use only the glosses for the synsets having that POS. Comparing the two methods experimentally allows to test the impact of sample phrases on textual representations. We have derived the third and fourth method by applying to the TD and TDS textual representations *negation propagation* [22], that consists in replacing all the terms that occur after a negation in a sentence with negated versions of the term [14] (e.g. in the sentence `This is not good`, the term `good` is converted to the term ¬`good`), thus yielding the $TD_{\neg}$ and $TDS_{\neg}$ methods. Figure 1.5 shows two textual representations created from it by using the TD and $TD_{\neg}$ methods.

1.4.5 Classification

We have classified terms by learning a classifier from the vectorial representations of the terms in $Tr_p^K \cup Tr_n^K$, and by then applying the

[14] Negation propagation has also been successfully used in [77] for sentiment classification of movie reviews.

```
Overview of noun unfortunate
The noun unfortunate has 1 sense (first 1 from tagged
texts)
1. unfortunate, unfortunate person -- (a person who suffers
misfortune)
Overview of adj unfortunate
The adj unfortunate has 3 senses (first 2 from tagged
texts)
1. unfortunate -- (not favored by fortune; marked or accompanied
by or resulting in ill fortune; ‘an unfortunate
turn of events’; ‘an unfortunate decision’; ‘unfortunate
investments’; ‘an unfortunate night for all concerned’)
2. inauspicious, unfortunate -- (not auspicious; boding
ill)
3. unfortunate -- (unsuitable or regrettable; ‘an unfortunate
choice of words’; ‘an unfortunate speech’)
```

Figure 1.4: WordNet output for the term `unfortunate`.

TD **representation** :

`unfortunate unfortunate person a person who suffers misfortune not favored by fortune marked or accompanied by or resulting in ill fortune inauspicious not auspicious boding ill unsuitable or regrettable`

$TD_\neg$ **representation** :

`unfortunate unfortunate person a person who suffers misfortune ¬favored ¬by ¬fortune marked or accompanied by or resulting in ill fortune inauspicious ¬auspicious boding ill unsuitable or regrettable`

Figure 1.5: TD and $TD_\neg$ representations for the term `unfortunate`.

resulting binary classifier (Positive vs. Negative) to the test terms. We have obtained vectorial representations for the terms from their textual representations by performing stop word removal and weighting by cosine-normalized $tfidf$; we have performed no stemming.

The learning algorithms we have tested are the naive Bayesian learner using the multinomial model (NB) [66], support vector machines using linear kernels [51], and the PrTFIDF probabilistic version of the Rocchio learner [50][15]. The use of the PrTFIDF learner is uncommon, but other works have reported good results with it (e.g. [82, 83]).

1.5 Results

The various combinations of choices of seed sets, expansion method (also considering the variable number of expansion steps steps), method for the creation of textual representations, and classification algorithm, resulted in several thousands different experiments. Therefore, in the following we only report the results we have obtained with the best-performing combinations.

Table 1.4 shows the accuracy obtained using the base seed sets (Tur and Kam) with no expansion and the NB classifier. The accuracy is still relatively low because of the small size of the training set, but for the KA term set the result obtained using $TDS_\neg$ representations is already better than the best accuracy reported in [53] on the same term set.

Table 1.4 shows an average 4.4% increase (with standard deviation $\sigma = 1.14$) in accuracy in using TDS representations versus TD ones, and an average 5.7% increase ($\sigma = 1.73$) by using representations obtained with negation propagation versus ones in which this has not been used. We have noted this trend also across all other experiments: the best performance, keeping all other parameters fixed, is always obtained using $TDS_\neg$ representations. For this reason in the rest of the paper we only report results obtained used the $TDS_\neg$ method.

Applying expansion methods to seed sets improves results just after a few iterations. Figure 1.6 illustrates the accuracy values obtained in the classification of the TL term set by applying expansion functions to the

[15] The naive Bayesian and PrTFIDF learners we have used are from McCallum's *Bow* package (`http://www-2.cs.cmu.edu/~mccallum/bow/`), while the SVM learner we have used is version 6.01 of Joachims' SVM^{light} (`http://svmlight.joachims.org/`).

Table 1.4: Accuracy (%) in classification using the base seed sets (with no expansion), the NB learner and various textual representations.

Seed set	Textual representation	TL	KA	HM
Kam	TD	48.47	53.41	50.01
Kam	TDS	52.47	54.63	53.69
Kam	$TD_{\neg}$	50.53	55.84	56.14
Kam	$TDS_{\neg}$	**53.81**	**58.55**	**58.76**
Tur	TD	57.86	64.49	56.91
Tur	TDS	59.56	65.10	61.06
Tur	$TD_{\neg}$	59.03	66.92	62.61
Tur	$TDS_{\neg}$	**61.18**	**68.53**	**65.49**

Kam seed set, using the various lexical relations or combinations thereof listed in Section 1.4.3. The $Hyper$ relation is not shown because it has always performed worse than with no expansion at all; a possible reason for this is that hypernymy, expressing the relation "is a kind of", very often connects (positively or negatively) oriented terms to non-oriented terms (e.g. `quality` is a hypernym of both `good` and `bad`).

Figure 1.6 also shows that the restriction to adjectives of the lexical relations (e.g. $Syn(J)$, $Ant^D(J)$, $Ant^I(J)$) produces better results than using the same relation without restriction on POS (e.g. $Syn(*)$, $Ant^D(*)$, $Ant^I(*)$). The average increase in accuracy obtained by bounding the lexical relations to adjectives versus not bounding them, measured across all comparable experiments, amounts to 2.88% ($\sigma = 1.76$). A likely explanation of this fact is that many word senses associated with POSs other than adjective are not oriented, even if other "adjective" senses of the same term are oriented (e.g. the noun `good`, in the sense of "product", has no orientation). This means that, when used in the expansion and in the generation of textual representations, these senses add "noise" to the data, which decreases accuracy. For instance, if no restriction on POS is enforced, expanding the adjective `good` through the synonymy relation will add the synonyms of the *noun* `good` (e.g. `product`) to Tr_p^K; and using the glosses for the "noun" senses of `good` will likely generate noisy representations.

Looking at the number of terms contained in the expanded sets after applying all possible iterations, we have, using the *Kam* seed set, 22,785 terms for $Syn(*)$, 14,237 for $Syn(J)$, 6,727 for $Ant^D(*)$, 6,021 for $Ant^D(J)$, 14,100 for $Ant^I(*)$, 13,400 for $Ant^I(J)$, 26,137 for $Syn(*) \cup Ant^I(*)$, and 16,686 for $Syn(J) \cup Ant^I(J)$. Expansions based on the *Tur* seed set are similar to those obtained using the *Kam* seed set, probably because of the close lexical relations occurring between the seven positive/negative terms. Across all the experiments, the average difference in accuracy between using the *Tur* seed set or the *Kam* seed set is about 2.55% in favour of the first ($\sigma = 3.03$), but if we restrict our attention to the 100 best-performing combinations we find no relevant difference (0.08% in favour of *Kam*, $\sigma = 0.43$).

Figure 1.6 shows that the best-performing relations are the simple $Syn(J)$ and $Ant^I(J)$ relations, and the combined relations $Syn(J) \cup Ant^I(J)$, $Syn(J) \cup Ant^D(J)$; these results are confirmed by all the experiments, across all learners, seed sets, and test sets.

Tables 1.5, 1.6 and 1.7 show the best results obtained on each seed set (*Tur* and *Kam*) on the HM, TL and KA test sets, respectively, indicating the learner used, the expansion method and the number of iterations applied, and comparing our results with the results obtained by previous works on the same test sets [44, 53, 92].

On the HM test set (Table 1.5) the best results are obtained with SVMs (87.38% accuracy), using the *Kam* seed set and the $Syn(J) \cup Ant^I(J)$ relation. Our best performance is 0.3% better than the best published result [92] and 12% better than the result of [44] on this dataset.

On the TL test set (Table 1.6) the best results are obtained with the *PrTFIDF* learner (83.09%) using the *Kam* seed set and the $Syn(J) \cup Ant^I(J)$ relation, thus confirming the results on the HM term set. Our best performance is 0.3% better than the only published result on this dataset [92].

On the KA test set (Table 1.7) the best results are obtained with SVMs (88.05%), again using the *Kam* seed set and the $Syn(J) \cup Ant^I(J)$ relation, again confirming the results on the TL and HM term sets. Our best performance is 31% better than the only published result on this dataset [53].

In a final experiment we have applied again the best-performing combinations, this time using textual representations extracted from the Merriam-Webster on-line dictionary (see Section 1.4.4) instead of

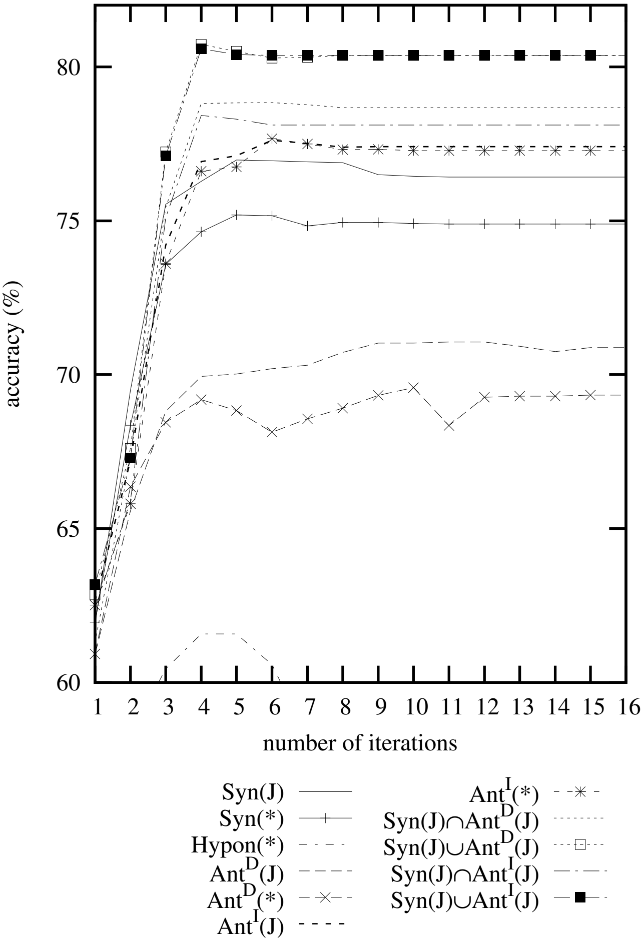


Figure 1.6: Accuracy in the classification (*NB* classifier) of the TL term set, using various lexical relations to expand the *Kam* seed set.

WordNet. We have obtained accuracies of 83.71%, 79.78%, and 85.44% on the HM, TL, and KA test sets, thus showing that it is possible to obtain acceptable results also by using resources other than WordNet.

In our comparisons with previously published methods we note that, while improvements with respect to the methods of [44, 53] have been dramatic, the improvements with respect to the method of [92] have been marginal. However, compared to the method of [92], ours is much less data-intensive: in our best-performing experiment on the TL term set we used an amount of data (consisting of the glosses of our terms) roughly 200,000 times smaller than the amount of data (consisting of the documents from which to extract co-occurrence data) required by the best-performing experiment of [92] (about half a million vs. about 100 billion word occurrences) on the same term set. The time required by our method for a complete run, from the iterative expansion of seed sets to the creation of textual representations, their indexing and classification, is about 30 minutes, while the best-performing run of [92] required about 70 hours. In an experiment using a volume of data only 20 times the size of ours (10 million word occurrences), [92] obtained accuracy values 22% inferior to ours (65.27% vs. 83.09%), and at the price of using the time-consuming PMI-LSA method. We should also mention that we bootstrap from a smaller seed set than [92], actually a subset of it containing only 1+1 seed terms instead of 7+7. Additionally, we should mention that our results are also fully reproducible. This is not true of the results of [92], due (i) to the fluctuations of Web content, and (ii) to the fact that the query language of the search engine used for those experiments (AltaVista) does not allow the use of the NEAR operator any longer.

1.6 Conclusions

In this chapter we have faced the problem of determining the orientation of subjective terms. We have presented a novel method for determining the orientation of subjective terms. The method is based on semi-supervised learning applied to term representations obtained by using term glosses from a freely available machine-readable dictionary. When tested on all the publicly available corpora for this task, this method has outperformed all the published methods, although the best-performing known method is beaten only by a small margin [92]. This result is valu-

Table 1.5: Best results in classification of the HM term set.

Method	Seed set	Expansion method	# of iterations (K)	Acc. (%)
[44]	–	–	–	78.08
SVM	Kam	$Syn(J) \cup Ant^I(J)$	8	**87.38**
$PrTFIDF$	Kam	$Syn(J) \cup Ant^D(J)$	4	84.73
NB	Kam	$Syn(J) \cup Ant^I(J)$	6	84.28
[92]	Tur	–	–	87.13
SVM	Tur	$Syn(J) \cup Ant^D(J)$	7	87.21
$PrTFIDF$	Tur	$Syn(J) \cup Ant^D(J)$	4	85.40
NB	Tur	$Syn(J) \cup Ant^D(J)$	5	84.73

Table 1.6: Best results in the classification of the TL term set.

Method	Seed set	Expansion method	# of iterations (K)	Acc. (%)
$PrTFIDF$	Kam	$Syn(J) \cup Ant^I(J)$	4	**83.09**
SVM	Kam	$Syn(J) \cup Ant^D(J)$	6	81.41
NB	Kam	$Syn(J) \cup Ant^D(J)$	4	80.73
[92]	Tur	–	–	82.84
$PrTFIDF$	Tur	$Syn(J) \cup Ant^I(J)$	3	82.20
SVM	Tur	$Syn(J) \cup Ant^I(J)$	9	81.91
NB	Tur	$Syn(J) \cup Ant^D(J)$	3	80.73

Table 1.7: Best results in the classification of the KA term set.

Method	Seed set	Expansion method	# of iterations (K)	Acc. (%)
[53]	Kam	–	–	67.32
SVM	Kam	$Syn(J) \cup Ant^I(J)$	4	**88.05**
$PrTFIDF$	Kam	$Syn(J) \cup Ant^D(J)$	8	87.59
NB	Kam	$Syn(J) \cup Ant^D(J)$	4	86.23
SVM	Tur	$Syn(J) \cup Ant^I(J)$	3	87.21
$PrTFIDF$	Tur	$Syn(J) \cup Ant^D(J)$	3	87.59
NB	Tur	$Syn(J) \cup Ant^D(J)$	7	86.38

able notwithstanding this small margin, since it was obtained with only 1 training term per category, and with a method $O(10^5)$ times less data-intensive and $O(10^2)$ times less computation-intensive than the method of [92].

Chapter 2

Determining Term Subjectivity and Term Orientation

Abstract

We contend that the plain determination of the orientation of terms, explored in the previous chapter, is not a realistic problem, since it starts from the non-realistic assumption that we already know whether a term is subjective or not; this would imply that a linguistic resource that marks terms as "subjective" or "objective" is available, which is usually not the case. In this chapter we confront the task of deciding whether a given term has a positive connotation, or a negative connotation, *or has no subjective connotation at all*; this problem thus subsumes the problem of determining subjectivity *and* the problem of determining orientation. We tackle this problem by testing three different variants of the semi-supervised term classification method previously proposed for orientation detection. Results show that determining subjectivity *and* orientation is a much harder problem than determining orientation alone.

2.1 Introduction

Implicit in most works dealing with term orientation is the assumption that, for many languages for which one would like to perform opinion mining, there is no available lexical resource where terms are tagged as having either a Positive or a Negative connotation, and that in the absence of such a resource the only available route is to generate such a resource automatically.

However, we think this approach lacks realism, since it is also true that, for the very same languages, there is no available lexical resource where terms are tagged as having either a Subjective or an Objective connotation. Thus, the availability of an algorithm that tags Subjective terms as being either Positive or Negative is of little help, since determining if a term is Subjective is itself non-trivial.

The need for a method that automatically generates a lexical resource on the subjectivity and orientation dimensions of opinion is also more relevant when observing the work of Bethard et al. [9], which report interesting results that show how automatically generated lexical resources, although inaccurate respect to human compiled ones, produce better results because of their larger *coverage* of the language.

In this chapter we confront the task of determining whether a given term has a Positive connotation (e.g. `delicious`, `nice`), or a Negative connotation (e.g. `disturbing`, `bogus`), or has instead no Subjective connotation at all (e.g. `white`, `triangular`); this problem thus subsumes the problem of deciding between Subjective and Objective *and* the problem of deciding between Positive and Negative.

We tackle this problem by testing three different variants of the semi-supervised method for orientation detection proposed in Chapter 1. Our results show that determining subjectivity *and* orientation is a much harder problem than determining orientation alone.

2.1.1 Chapter outline

The rest of the chapter is structured as follows. Section 2.2 reviews related work dealing with term orientation and/or subjectivity detection. Section 2.3 reviews the semi-supervised method for orientation detection presented in Chapter 1, pointing out which are the relevant point to be modified to work on the new task. Section 2.4 describes in detail three different variants we propose for determining, at the same time, subjectivity *and* orientation, and also describes the general setup of our experiments. In Section 2.5 we discuss the results we have obtained. Section 2.6 concludes.

2.2 Related work

2.2.1 Determining term orientation

For a discussion of literature related to the problem of determining term orientation we point the reader to Section 1.2 of Chapter 1.

Among those work, we are interested to recall that Kim and Hovy [54] have reported an evaluation of human inter-coder agreement on the task of subjectivity recognition, on adjective and verbs. We will compare this evaluation with our results in Section 2.5.

2.2.2 Riloff et al.

Riloff et al. [81] have developed a method to determine whether a term has a Subjective or an Objective connotation, based on bootstrapping algorithms. The method identifies patterns for the extraction of subjective *nouns* from text, bootstrapping from a seed set of 20 strongly subjective terms selected by the authors. terms that the authors judge to be strongly subjective and have found to have high frequency in the text collection from which the subjective nouns must be extracted.

The results of this method are not easy to compare with the ones we present in this paper because of the different evaluation methodologies. While we adopt the evaluation methodology used in all of the papers reviewed so far (i.e. checking how good our system is at replicating an existing, independently motivated lexical resource), – an evaluation methodology standard in the field of information retrieval – the authors of [81] have not tested their method on an independently identified set of labeled terms, but on the set of terms that the algorithm itself extracts. This evaluation methodology only allows to test precision, and not accuracy *tout court*, since no quantification can be made of false negatives (i.e. the subjective terms that the algorithm should have spotted but has not spotted). In Section 2.5 this will prevent us from drawing comparisons between this method and our own.

2.2.3 Baroni and Vegnaduzzo

Baroni and Vegnaduzzo [7] have applied the PMI method, first used by Turney and Littman [92] (see Chapter 1, Section 1.2.2) to determine term orientation, to determine term subjectivity. Their method uses a small set S_s of 35 adjectives, marked as subjective by human judges, to assign a subjectivity score to each adjective to be classified. Therefore, their method, unlike our own, does not *classify* terms (i.e. take firm classification decisions), but *ranks* them according to a subjectivity score, on which they evaluate precision at various level of recall.

Unfortunately we cannot compare our method with the methods of [7, 81] because of the specific evaluation measures adopted in these works, as mentioned above, and also because they were not evaluated on publicly available datasets.

2.3 Determining term subjectivity and term orientation by semi-supervised learning

The method we use in this chapter for determining term subjectivity and term orientation is a variant of the method proposed in Chapter 1 for determining term orientation alone. We recall here the main components of the method, analyzing which modifications are required to support the new task.

The process for determining the orientation of terms is composed of the following steps:

1. Two seed sets Tr_p^0 and Tr_n^0, of terms that are representative of the two categories Positive and Negative, are provided as input.

2. Lexical relations (e.g. synonymy) from a thesaurus, or online dictionary, are used in order to find new terms that will also be considered representative of the two categories because of their relation with the terms contained in Tr_p^0 and Tr_n^0. This process can be iterated K times, using as input at every iteration k the result of the previous one $k-1$. The new terms, once added to the original ones, yield two new, richer sets Tr_p^K and Tr_n^K of terms; together they form the training set for the learning phase of Step 4.

3. For each term t_i in $Tr_p^K \cup Tr_n^K$ or in the test set (i.e. the set of terms to be classified), a textual representation of t_i is generated by collating all the glosses of t_i as found in a machine-readable dictionary[1]. Each such representation is converted into vectorial form by standard text indexing techniques.

[1] In general a term t_i may have more than one gloss, since it may have more than one sense; dictionaries normally associate one gloss to each sense.

4. A binary text classifier is trained on the terms in $Tr_p^K \cup Tr_n^K$ and then applied to the terms in the test set.

The new task consists in a *single label* classification task of terms over the three categories Positive, Negative and Objective. This variation have non-trivial effects on the method, which are listed here, while details on how the variants to the method have been realized to handle such effects are described in the Section 2.4.

Respect to the Step 1 the new task requires *three* seed sets, Tr_p^0, Tr_n^0 and Tr_o^0. While it is relatively easy, as we have shown in Chapter 1, to identify a small set of terms that are representative of the two concepts of positivity and negativity, it is not intuitive to figure out how a small set of terms could represent the concept of objectivity, given its heterogeneousness. In fact the Objective category is a "container" for terms that *have not* the property of being either Positive or Negative, but it does not describe any other property the terms may *have*. For example, the terms `triangular`, `chocolate`, `walk` are all Objective, but otherwise completely unrelated one from the other. Section 2.4.2 describes a possible solution to this issue.

Again in Step 2 we have to face the problem of the "container" nature of the Objective category. For Positive and Negative it is relatively intuitive that the synonymy and antonymy relations could be used to find other terms with the same (or opposite) orientation. For the Objective category the intuition seems to be that most of the lexical relations could be used to find new objective terms starting from a source term, but it is difficult to think of a lexical relation that connects objective terms with oriented ones. Section 2.4.2 describes how the expansion phase have been realized in experiments.

The phase of construction of textual representations of terms in Step 3 does not present any aspect that is related with the number of categories involved in the learning or classification process, thus it could be left unchanged.

The last Step 4 of training and classification does present instead the most relevant differences. In the original task on term orientation, the classification on the two mutually-exclusive categories Positive and Negative the choice of learning a single binary classifier is an obvious solution. In the new configuration with *three* mutually-exclusive categories, a single binary classifier cannot cover all the possible results of a classification, and the use of a combination of many binary classifiers is required. However there are many possible approaches in building and combining some binary classifier to obtain a *1-of-3* classifier. For example, one can consider the union of the Positive and Negative categories as a Subjective category, and then learn two binary classifiers, one on the Subjective/Objective dichotomy and one on the Positive/Negative. At classification time a term will be first classified by the Subjective/Objective classifier and then, if classified as Subjective, will be classified by the Positive/Negative classifier. This is just one of the three variants we explored in our experiments, which are all described in details in Section 2.4.3

2.4 Experiments

2.4.1 Test sets

The benchmark (i.e. test set) we use for our experiments is derived again from the General Inquirer (GI) lexicon [86], like in Chapter 1. In this case we have to define also a set of Objective terms, in addition to the Positive and Negative ones. For the latter ones we used the same list of 1,612/1,982 terms obtained by two original GI sets of 1,915/2,291 terms after removing 17 terms appearing in both categories (e.g. `deal`) and reducing all the multiple entries of the same term in a category, caused by multiple senses, to a single entry. For the Objective category, we likewise take all the 7,582 GI terms that are not labeled as either Positive or Negative, as being (implicitly) labeled as Objective, and reduce them to 5,009 terms after combining multiple entries of the same term, caused by multiple senses, to a single entry. This seems a sensible assumption, since we can assume that, if a given term were not Objective, the lexicographers who have created GI would have classified it as either Positive or Negative.

The effectiveness of our classifiers will thus be evaluated in terms of their ability to assign the total 8,605 GI terms to the correct category among Positive, Negative, and Objective[2].

[2] This labeled term set is available for download at `http://patty.isti.cnr.it/~esuli/software/SentiGI.tgz`.

2.4.2 Seed sets and training sets

Similarly to Chapter 1, our training set is obtained by expanding initial seed sets by means of WordNet lexical relations. The main difference is that our training set is now the union of *three* sets of training terms $Tr = Tr_p^K \cup Tr_n^K \cup Tr_o^K$ obtained by expanding, through K iterations, three seed sets Tr_p^0, Tr_n^0, Tr_o^0, one for each of the categories Positive, Negative, and Objective, respectively.

Concerning categories Positive and Negative, we have used the seed sets, expansion policy, and number of iterations, that have performed best in the experiments on terms orientation (see Chapter 1, Section 1.5), i.e. the seed sets $Tr_p^0 = \{\texttt{good}\}$ and $Tr_n^0 = \{\texttt{bad}\}$ (which have been called the *Kam* seed sets, since they were originally used by Kamps et al. in [53]) expanded by using the union of the WordNet relations of synonymy and indirect antonymy, restricting the relations only to terms with the same POS of the original terms (i.e. adjectives), for a total of $K = 4$ iterations. The final expanded sets contain 6,053 Positive terms and 6,874 Negative terms.

Concerning the category Objective, the process we have followed is similar, but with a few key differences. These are motivated by the fact that the Objective category coincides with the complement of the union of Positive and Negative; therefore, Objective terms are more varied and diverse in meaning than the terms in the other two categories, as already described in Section 2.3. To obtain a representative expanded set Tr_o^K, we have chosen the seed set $Tr_o^0 = \{\texttt{entity}\}$ and we have expanded it by using, along with synonymy and antonymy, the WordNet relation of hyponymy (e.g. `vehicle / car`), and without imposing the restriction that the two related terms must have the same POS. These choices are strictly related to each other: the term `entity` is the root term of the largest generalization hierarchy in WordNet, with more than 40,000 terms [24], thus allowing to reach a very large number of terms by using the hyponymy relation[3]. Moreover, it seems reasonable to assume that terms that refer to *entities* are likely to have an "objective" nature, and that hyponyms (and also synonyms and antonyms) of an objective term are also objective. Note that, at each iteration k, before adding a given term t to Tr_o^k we check if it already belongs to either Tr_p^K or Tr_n^K; if it does the term is not added to Tr_o^k and is discarded from consideration

[3] The largest connected component for the synonymy relation consists instead of only 10,922 names, as reported in [53].

(i.e. is expanded no further). We experiment with two different choices for the Tr_o^K set, corresponding to the sets generated in $K = 3$ and $K = 4$ iterations, respectively; this yields sets Tr_o^3 and Tr_o^4 consisting of 8,353 and 33,870 training terms, respectively.

It is interesting to observe that if the final sets Tr_p^K, Tr_n^K, Tr_o^K resulting from expansion (with $K = 4$) are used to directly classify GI terms, the accuracy in classification is 58.20%, with a coverage of 82.32% of the total number of GI terms, which drops to 47.91% if calculated on the whole GI. These values can be used as a non-trivial baseline to evaluate our results.

2.4.3 Learning approaches and evaluation measures

We experiment with three "philosophically" different learning approaches to the problem of making a single-label classification over the Positive, Negative, and Objective categories.

Approach I is a two-stage method which consists in learning two binary classifiers (see Section 2.4.3: the first classifier places terms into either Subjective or Objective, while the second classifier places terms that have been classified as Subjective by the first classifier into either Positive or Negative. In the training phase, the terms in $Tr_p^K \cup Tr_n^K$ are used as training examples of category Subjective.

Approach II is again based on learning two binary classifiers. Here, one of them must discriminate between terms that belong to the Positive category and ones that belong to its complement (**not** Positive), while the other must discriminate between terms that belong to the Negative category and ones that belong to its complement (**not** Negative). Terms that have been classified *both* into Positive by the former classifier and into (**not** Negative) by the latter are deemed to be positive, and terms that have been classified *both* into (**not** Positive) by the former classifier and into Negative by the latter are deemed to be negative. The terms that have been classified (i) into both (**not** Positive) and (**not** Negative), or (ii) into both Positive and Negative, are taken to be Objective. The choice we apply when condition (ii) happens, is motivated by the structure of our test set (see Section 2.4.1) that defines a *single-label* classification problem (i.e. to assign to a term one of three labels): when a term is classified as both Positive and Negative, is not possible to choose between one of the two category, thus leaving the third choice, Objective, as a way out. Another possible classification, proper of a *multilabel* classification problem, is to consider the term as both Positive and Negative (e.g. a term may have multiple meanings with different orientations[4]). In Section 2.5) we report on the frequency on which condition (ii) happens. In the training phase of Approach II, the terms in $Tr_n^K \cup Tr_o^K$ are used as training examples of category (**not** Positive), and the terms in $Tr_p^K \cup Tr_o^K$ are used as training examples of category (**not** Negative).

Approach III consists instead in viewing Positive, Negative, and Objective as three categories with equal status, and in learning a ternary classifier that classifies each term into exactly one among the three categories.

There are several differences among these three approaches. A first difference, of a conceptual nature, is that only Approaches I and III view Objective as a category, or concept, in its own right, while Approach II views objectivity as a nonexistent entity, i.e. as the "absence of subjectivity" (in fact, in Approach II the training examples of Objective are only used as training examples of the *complements* of Positive and Negative). A second difference, more of a technological nature, is that Approaches I and II are based on standard binary classification technology, while Approach III requires "multiclass" (i.e. 1-of-m) classification. As a consequence, while for the former we use well-known learners for binary classification (the naive Bayesian learner using the multinomial model [66], support vector machines using linear kernels [51], the Rocchio learner, and its PrTFIDF probabilistic version [50]), for Approach III we use their multiclass versions[5].

Before running our learners we make a pass of feature selection, with the intent of retaining only those features that are good at discriminating our categories, while discarding those which are not. Feature selection is implemented by scoring each feature f_k (i.e. each term that occurs in the glosses of at least one training term) by means of the *mutual information* (MI) function, defined as [6]

[4] We do not experimented on this because is out of the focus of this Chapter, but this will be the main topic of Chapters 4 and 5.

[5] The naive Bayesian, Rocchio, and PrTFIDF learners we have used are from Andrew McCallum's *Bow* package (`http://www-2.cs.cmu.edu/~mccallum/bow/`), while the SVMs learner we have used is Thorsten Joachims' SVM^{light} (`http://svmlight.joachims.org/`), version 6.01. Both packages allow the respective learners to be run in "multiclass" fashion.

[6] Mutual information is a function from information theory which is also known as *information gain*, and is sometimes given in the equivalent form $MI(f_k, c_i) = H(c_i) - H(c_i|t_k)$, where $H(X)$ is the *entropy* of X and $H(X|Y)$ is the *conditional*

$$MI(f_k) = \sum_{c \in \{c_1,\ldots,c_m\}, f \in \{f_k, \overline{f}_k\}} \Pr(f,c) \cdot \log \frac{\Pr(f,c)}{\Pr(f)\Pr(c)}$$

and discarding the $x\%$ features f_k that minimize it. We will call $x\%$ the *reduction factor*.

In categorization applications MI serves the purpose of measuring the discriminative power of a feature with respect to a set of classes $C = \{c_1, \ldots, c_m\}$, i.e. of evaluating the expected quality of the contribution that a feature will give to the categorization task.

Note that the set $\{c_1, \ldots, c_m\}$ from Equation 2.1 is interpreted differently in Approaches I to III, and always consistently with who the categories at stake are.

2.4.4 Evaluation measures

Since the task we aim to solve is manifold, we will evaluate our classifiers according to two evaluation measures:

- *SO-accuracy*, the accuracy of a classifier in separating Subjective from Objective, i.e. in deciding term subjectivity alone;
- *PNO-accuracy*, the accuracy of a classifier in discriminating among Positive, Negative, and Objective, i.e. in deciding both term orientation and subjectivity.

Note that we use instances of *accuracy*, i.e. the percentage of correct decisions out of the total of all classification decisions, and not F_1 [62]: this latter measure is appropriate for binary (i.e. n-of-m) classification contexts[7], while in our case we always have to pick exactly one category per term (i.e. ours is a 1-of-m context).

Note also that we do not present PN-accuracy results (i.e. orientation) given that this had been extensively discussed in Chapter 1.

2.5 Results

We present results obtained from running every combination of (i) the three approaches to classification described in Section 2.4.3, (ii) the four

entropy of Y given X [20, page 19].

[7] See also Chapter 3, Section 3.5.2.

Table 2.1: Average and best accuracy values over the four dimensions analyzed in the experiments.

Dimension	SO-accuracy		PNO-accuracy	
	Avg (σ)	Best	Avg (σ)	Best
Approach				
I	.635 (.020)	.668	.595 (.029)	.635
II	**.636** (.033)	**.676**	**.614** (.037)	**.660**
III	.635 (.036)	.674	.600 (.039)	.648
Learner				
NB	**.653** (.014)	.674	**.619** (.022)	.647
SVMs	.627 (.033)	.671	.601 (.037)	.658
Rocchio	.624 (.030)	.654	.585 (.033)	.616
PrTFIDF	.637 (.031)	**.676**	.606 (.042)	**.660**
TSR				
0%	.649 (.025)	**.676**	.619 (.027)	**.660**
50%	**.650** (.022)	.670	**.622** (.022)	.657
80%	.646 (.023)	.674	.621 (.021)	.647
90%	.642 (.024)	.667	.616 (.024)	.651
95%	.635 (.027)	.671	.606 (.031)	.658
99%	.612 (.036)	.661	.570 (.049)	.647
Tr_o^K *set*				
Tr_o^3	**.645** (.006)	**.676**	.608 (.007)	.658
Tr_o^4	.633 (.013)	.674	**.610** (.018)	**.660**

learners mentioned in the same section, (iii) five different reduction factors for feature selection (0%, 50%, 90%, 95%, 99%), and (iv) the two different training sets (Tr_o^3 and Tr_o^4) for Objective mentioned in Section 2.4.2. We discuss each of these four dimensions of the problem individually, for each one reporting results averaged across all the experiments we have run (see Table 2.1).

The first and most important observation is that, with respect to a pure term orientation task, accuracy drops significantly. In fact, the best SO-accuracy and the best PNO-accuracy results obtained across the 120 different experiments are .676 and .660, respectively (these were obtained by using Approach II with the PrTFIDF learner and no feature selection, with $Tr_o^K = Tr_o^3$ for the .676 SO-accuracy result and $Tr_o^K = Tr_o^4$ for the .660 PNO-accuracy result); this contrasts sharply with the accuracy obtained in Chapter 1 on discriminating Positive terms from

Table 2.2: Human inter-coder agreement values reported by Kim and Hovy [54].

Agreement measure	Adjectives (462)	Verbs (502)
	Hum1 vs Hum2	Hum2 vs Hum3
Strict	.762	.623
Lenient	.890	.851

Negative ones (where the best run obtained .830 accuracy), *on the same benchmarks and essentially the same algorithms.* Note that the trivial baseline, obtained by always picking the majority class, is .582; This suggests that good performance at orientation detection may not be a guarantee of good performance at subjectivity detection, quite evidently a harder (and, as we have suggested, more realistic) task.

This hypothesis is confirmed by an experiment performed by Kim and Hovy [54], on testing the agreement of two human coders at tagging words with the Positive, Negative, and Objective labels. The authors define two measures of such agreement: *strict* agreement, equivalent to our PNO-accuracy, and *lenient* agreement, which measures the accuracy at telling Negative against the rest. For any experiment, strict agreement values are then going to be, by definition, lower or equal than the corresponding lenient ones. The authors use two sets of 462 adjectives and 502 verbs, respectively, randomly extracted from the basic English word list for foreign students preparing for of the TOEFL test. The inter-coder agreement results (see Table 2.2) show a deterioration in agreement (from lenient to strict) of 16.77% for adjectives and 36.42% for verbs. Following this, we evaluated our best experiment according to these measures, and obtained a "strict" accuracy value of .660 and a "lenient" accuracy value of .821, with a relative deterioration of 24.39%, in line with Kim and Hovy's observation[8]. This confirms that determining subjectivity and orientation is a much harder task than determining orientation alone.

The second important observation is that there is very little variance in the results: across all 120 experiments, average SO-accuracy and PNO-accuracy results were .635 (with standard deviation $\sigma = .030$) and .603 ($\sigma = .036$), a mere 6.06% and 8.64% deterioration from the best results reported above. This seems to indicate that the levels of perfor-

[8] We observed this trend in all of our experiments.

mance obtained may be hard to improve upon, especially if working in a similar framework.

Let us analyze the individual dimensions of the problem. Concerning the three approaches to classification described in Section 2.4.3, Approach II outperforms the other two, but by an extremely narrow margin. As for the choice of learners, on average the best performer is NB, but again by a very small margin with respect to the others. On average, the best reduction factor for feature selection turns out to be 50%, but the performance drop we witness in approaching 99% (a dramatic reduction factor) is extremely graceful. As for the choice of Tr_o^K, we note that Tr_o^3 and Tr_o^4 elicit comparable levels of performance, with the former performing best at SO-accuracy and the latter performing best at PNO-accuracy.

An interesting observation on the learners we have used is that NB, PrTFIDF and SVMs, unlike Rocchio, generate classifiers that depend on $P(c_i)$, the *prior probabilities* of the classes, which are normally estimated as the proportion of training documents that belong to c_i. In many classification applications this is reasonable, as we may assume that the training data are sampled from the same distribution from which the test data are sampled, and that these proportions are thus indicative of the proportions that we are going to encounter in the test data. However, in our application this is not the case, since we do not have a "natural" sample of training terms. What we have is one human-labeled training term for each category in {Positive, Negative, Objective}, and as many machine-labeled terms as we deem reasonable to include, in possibly different numbers for the different categories; and we have no indication whatsoever as to what the "natural" proportions among the three might be. This means that the proportions of Positive, Negative, and Objective terms we decide to include in the training set will strongly bias the classification results if the learner is one of NB, PrTFIDF and SVMs. We may notice this by looking at Table 2.3, which shows the average proportion of test terms classified as Objective by each learner, depending on whether we have chosen Tr_o^K to coincide with Tr_o^3 or Tr_o^4; note that the former (resp. latter) choice means having roughly as many (resp. roughly five times as many) Objective training terms as there are Positive and Negative ones. Table 2.3 shows that, the more Objective training terms there are, the more test terms NB, PrTFIDF and (in particular) SVMs will classify as Objective; this is not true for Rocchio, which is

Table 2.3: Average proportion of test terms classified as Objective, for each learner and for each choice of the Tr_o^K set.

Learner	Tr_o^3	Tr_o^4	Variation
NB	.564 ($\sigma = .069$)	.693 (.069)	+23.0%
SVMs	.601 (.108)	.814 (.083)	+35.4%
Rocchio	.572 (.043)	.544 (.061)	**-4.8%**
PrTFIDF	.636 (.059)	.763 (.085)	+20.0%

basically unaffected by the variation in size of $T^K r_o$.

In our experiments, as is commonly done in supervised classification problems, $P(c)$ is estimated as the ratio of the number of examples belonging to c versus the total number of examples in the training set, following the hypothesis that the proportions between the categories in the train set is representative of a "real world" distribution; but in our case we can't guarantee this property because the proportions between categories depend mostly by the choice of the Objective set Tr_o^K, that is generated using a different method respect to Positive and Negative. Table 2.3 shows the average number of Objective classifications obtained using each learner, depending on the training set used. The first evidence is that, as discussed previously, the Rocchio classifier is not affected by the proportions between categories. The other three learners instead produce a greater number of Objective classifications as this category has a greater number of positive examples in the training set, with SVMs more sensible to the variation respect to NB and PrTFIDF.

Finally, we have measured the time required by each experiments, consisting in performing feature selection on the term glosses, training the classifier, and applying it to the test set. These timings do not include the time required to compute the expanded Tr_x^K sets (2 minutes for $K = 3$, 6 minutes for $K = 4$), and to extract and process WordNet glosses of training and test terms (32 minutes for the training set that uses Tr_o^3, 72 minutes for the one that uses Tr_o^4, and 14 minutes for the test set). The average times for the various learners and approaches are shown in Table 2.4. The first observation is that SVMs are more computationally demanding than the other learners, which require no more than two minutes for a complete run. The multiclass version of SVMs, used in Approach III, is the least efficient learner, ten times less efficient (on average) than the single-class version used in Approach I.

Table 2.4: Average time (sec) for a classification experiment (i.e. learning and classification) for each of the four learners, over the two Tr_o^K sets and the three Approaches used.

Learner	Approach		
	I	II	III
Tr_o^3			
NB	75 ($\sigma = 23.1$)	75 (25.5)	**56** (24.2)
SVMs	**222** (91.1)	335 (106.6)	1685 (264.0)
Rocchio	**63** (2.0)	76 (26.2)	64 (2.7)
PrTFIDF	67 (3.1)	75 (24.9)	**65** (3.3)
Tr_o^4			
NB	107 (32.3)	163 (30.4)	**94** (32.6)
SVMs	**516** (116.0)	853 (164.5)	5814 (994.7)
Rocchio	115 (23.0)	146 (48.5)	**85** (10.7)
PrTFIDF	105 (29.9)	154 (30.5)	**83** (30.3)

The other three learners have a similar behavior, that is mostly driven by the amount of data to be processed during the learning phase. In Approach III the training data are processed only once, which means this is the fastest approach. In Approach I the training data are fully processed once, to learn the Subjective vs. Objective classifier, and then the Positive and Negative training examples are re-processed to learn the Positive vs. Negative classifier. In Approach II the training data are processed twice, to build the two binary classifiers for the Positive and Negative categories, respectively, which means this is the slowest approach.

Respect to the special condition (ii) of the Approach II, described in Section 2.4.3, we measured it to happen relatively rarely for the classifiers based on SVMs (on the average, 0.5% of the times, $\sigma = 0.3$) and naive Bayes (1.9%, $\sigma = 0.6$) and relatively frequently when using classifiers based on PrTFIDF (7.6%, $\sigma = 6.4$) or Rocchio (14.9%, $\sigma = 9.9$).

2.6 Conclusions

We have presented a method for determining *both* term subjectivity *and* term orientation for opinion mining applications. This is a valuable advance with respect to the state of the art, since past work in this area had mostly confined to determining term orientation alone, a task that (as we have argued) has limited practical significance in itself, given the generalized absence of lexical resources that tag terms as being either Subjective or Objective. Our algorithms have tagged by orientation *and* subjectivity the entire General Inquirer lexicon, a complete general-purpose lexicon that is the *de facto* standard benchmark for researchers in this field. Our results thus constitute, for this task, the first baseline for other researchers to improve upon.

Unfortunately, our results have shown that an algorithm that had shown excellent, state-of-the-art performance in deciding term orientation (see Chapter 1, once modified for the purposes of deciding term subjectivity, performs more poorly. This has been shown by testing several variants of the basic algorithm, some of them involving radically different supervised learning policies. The results suggest that deciding term subjectivity is a substantially harder task that deciding term orientation alone.

Chapter 3

Determining Attitude Type and Force

Abstract

In previous chapters we have focused on recognizing the subjectivity and orientation of terms. The focus of this chapter is on two other dimensions which characterize the expression of opinions, the *attitude* and the *force*. The attitude is a fine-grained property which identifies the type of appraisal being expressed, e.g. `beautiful` expresses appreciation of an object's quality, while `evil` expresses a negative judgment of social behavior. The force attribute identifies the intensity of the appraisal, e.g. the intensity-increasing scale of terms: `tasteful`, `delicious`, `luscious`. In this chapter we describe the application of our term classification method to the automatic determination of *attitude* and *force* of terms.

3.1 Introduction

Most of the literature on OM is focused on recognizing subjective expressions, and possibly their orientation, in text. Such information have an immediate use in many applications, e.g. comparing products, analyzing products' features, survey people's opinions. However, subjective expressions have not just the property of been positive or negative. Subjective expressions may express *appraisal* with different types of *attitudes*, e.g. the term `elegant` expresses an *appreciation* for an object's quality, the term `honest` expresses a *judgment* for a moral quality, while the term `happy` expresses an affective state. Also the intensity of the appraisal expression may vary, e.g. from `tasteful`, to `delicious`, to `luscious`.

Some applications, based on the use of hand-built lexical resources that define attitude types for a relatively small set of terms, have already shown the positive effect of using attitude-related information on the OM tasks [88, 94]. The multi-dimensional information contained in such resources allows to draw more subtle distinctions in evaluative texts than just classifying terms as Positive or Negative.

Unfortunately, manual development of this kind of resources has a high cost, typically higher than developing just a subjectivity lexicon, given the higher complexity of the taxonomy (see Figure 3.1). In fact in the work of Taboada and Grieve [88] the attitude lexicon is composed by just fifty adjectives, and in Whitelaw et al. [94] the lexicon, which is built semi-automatically and then manually checked, is composed by 1,847 terms (see Section 3.3).

The purpose of the work described in this chapter is to explore the use of our term classification method, used in the previous chapters to automatically determine the orientation and subjectivity of terms, on the much more fine-grained taxonomy of *attitude types*, which identify the type of appraisal expressed by a term, and the *force* of the appraisal expression.

The definition of attitude types, and force levels, used in this chapter is based on the framework of Martin and White's [65] Appraisal Theory, developed for the manual analysis of evaluative language. This framework assigns several sentiment-related features to relevant lexical items, including *orientation*, *attitude type* (whether Affect, Appreciation of inherent qualities, or Judgment of social interactions), and *force* of opinion expressed (High, Median, or Low).

3.1.1 Chapter outline

After a brief overview of relevant aspects of Appraisal Theory (Section 3.2), Section 3.3 reports about works that have already investigated other dimensions of the opinion-related semantic of terms than the simple subjectivity or orientation. We then describe our method for automatic classification of terms by attitude type and force (Section 3.4). The experimental setup is described in Section 3.5. Results are reported is Section 3.6 and conclusions in Section 3.7.

3.2 Appraisal Theory

Appraisal Theory is a systemic-functional approach to analyzing how subjective language is used to adopt or express an attitude of some kind towards some target [65].

Martin and White [65] model appraisal as comprising three main linguistic systems: "Attitude", which distinguishes different kinds of attitudes that can be expressed (including Attitude Type and Orientation); "Amplification", which enables strengthening or weakening such expression (including Force and Focus); and "Engagement", which conveys different possible degrees of commitment to the opinion expressed (including identification and relation of the speaker/writer to the source of an attributed evaluation).

Previous applications of Appraisal Theory to sentiment analysis [88, 94] have focused on three key components:

Attitude Type: specifies the type of appraisal being expressed as one of Appreciation, Affect, or Judgment (with further sub-typing possible). Affect refers to a personal emotional state (e.g., `happy`, `angry`), and is the most explicitly subjective type of appraisal. The other two options differentiate between the Appreciation of 'intrinsic' object properties (e.g., `slender`, `ugly`) and social Judgment (e.g., `heroic`, `idiotic`). Figure 3.1 gives a detailed view of the Attitude Type taxonomy, together with illustrative adjectives. In general, attitude type may be expressed through nouns (e.g., `triumph`, `catastrophe`) and verbs (e.g., `love`, `hate`), as well as adjectives.

Force: describes the intensity of the appraisal being expressed. Force may be realized via modifiers such as `very` (increased force) or `slightly` (decreased force), or may be realized lexically in a head word, e.g., `wonderful` vs. `great` vs. `good`.

Orientation: determines whether the appraisal is Positive or Negative.

```
Attitude Type
└─Appreciation
│ ├─Composition
│ │ ├─Balance: consistent, discordant, ...
│ │ └─Complexity: elaborate, convoluted, ...
│ ├─Reaction
│ │ ├─Impact: amazing, compelling, dull, ...
│ │ └─Quality:  beautiful, elegant, hideous, ...
│ └─Valuation:  innovative, profound, inferior, ...
├─Affect: happy, joyful, furious, ...
└─Judgment
  └─Social Esteem
  │ ├─Capacity: clever, competent, immature, ...
  │ ├─Tenacity: brave, hard-working, foolhardy, ...
  │ └─Normality: famous, lucky, obscure, ...
  └─Social Sanction
    ├─Propriety: generous, virtuous, corrupt, ...
    └─Veracity: honest, sincere, sneaky, ...
```

Figure 3.1: Options in the "Attitude Type" taxonomy, with examples of appraisal adjectives from the base lexicon described in Section 3.5.1.

3.3 Related work

Little research to date has applied such schemes in a computational context.

3.3.1 Taboada and Grieve

Taboada and Grieve [88] were among the first to consider the use of Appraisal Theory's semantic taxonomies for sentiment analysis. They have used a small lexicon of fifty adjectives manually classified for top-level attitude type, expanded by a technique based on pointwise mutual information (PMI) [92]. This lexicon has been used to assign a score to a corpus of documents, composed of product, book, and movie reviews, on each of the following three appraisal categories: Affect, Appreciation and Judgment. Authors' suggestion is that this scoring could help recognition of the type of review because, for example, products reviews typically have a high Appreciation score, while books reviews typically have a high Affect score. Another suggested application is to filter documents with respect to the highest rated attitude type: e.g. a customer reading hotel reviews could be interested to the elegance of the room (typically reported by Appreciation expressions) or the quality of the

Table 3.1: Whitelaw et al. [94] results, on the movie review corpus [77].

Features	Number of Features	Accuracy
BoW	48,314	87.0%
BoW + Attitude + Orientation	49,911	**90.2%**
Original BoW [77]	N/A	87.2%

service (typically reported by Judgment expressions).

3.3.2 Whitelaw et al.

Whitelaw et al. [94] have developed a method for using a structured lexicon of appraisal adjectives and modifiers to perform chunking and analysis of multi-word adjectival groups expressing appraisal, e.g. the expression `not very friendly` is analyzed as having Positive orientation, Propriety attitude type, and Low force. Their experimental results have shown that using such "appraisal groups" to generate additional features in the representation of documents from the *movie reviews corpus* from Pang et al. [77] produces a relevant accuracy improvement in document-level orientation detection (see Table 3.1). The lexicon they have used is built semi-automatically and then manually checked and is composed by 1,847 terms. We have used this lexicon in our experiments to evaluate our method (see Section 3.5.1).

3.3.3 Related models

The semantic features we analyze in this chapter are also related to other analysis of term "value" or "sentiment" in the literature. Osgood's [73] Theory of Semantic Differentiation delineated three dimensions of affective meaning: "evaluative", i.e., Orientation; "potency", referring to the strength of feeling expressed; and "activity", referring to how active or passive an evaluation is. This was the basis for Kamps and Marx's [52] analysis of affective meaning in WordNet. Mullen and Collier [70] have estimated values for Osgood's three dimensions for adjectives in WordNet, by comparing path lengths to appropriate pairs of anchor words (such as `good` and `bad`) in WordNet's synonymy graph, using document-level averages of these values as input to SVMs for sentiment classification.

Another relevant set of lexical attributes is given by the Lasswell Value Dictionary, as applied in the General Inquirer [86]. The purpose there is to classify words as relating to various basic "values", such as wealth, power, respect, rectitude, skill, enlightenment, affection, and wellbeing. Some of these have parallels in Appraisal Theory (for example "rectitude", which is similar to the attitude type of Social Sanction), while other Lasswell categories, such as "wealth" or "enlightenment" appear unrelated to any Attitude Type.

3.4 Learning attitude type and force

This work explores how a lexicon such as that used in [94] can be learned in a fully automatic fashion, concentrating on assigning the correct *attitude type* and *force* to terms. We have actually used the lexicon of Whitelaw et al. [94] as benchmark and also as input for classification method, as detailed in Section 3.5.1. The following sections describe the characteristics of the classification problems we have faced and how they have been solved with our method.

3.4.1 Seed sets expansion

Regarding the step automatic expansion of training sets of our method, in both cases of attitude and force, the algorithm takes in input n seed sets $Tr^0 = \{Tr^0_1, \ldots, Tr^0_n\}$ (with n defined by the current classification problem under investigation) and expands them into the final n training sets $Tr^K = \{Tr^K_1, \ldots, Tr^K_n\}$ after K iteration steps. Differently from the expansion process on the orientation task, synonyms *and* antonyms of a training term are added to the training set of the *same* class, following the intuition that antonyms will typically differ in *orientation* but neither in *attitude type* nor in *force*. For example, the category Balance in our lexicon includes terms such as `consistent` and `discordant`, while the category Tenacity includes terms such as `brave` and `foolhardy`. Moreover, only for attitude type, the inclusion of a term in the training set of a category i does not exclude its successive inclusion in the training set of another category, given the multi-label nature of the task.

3.4.2 Attitude classification

Determining attitude type consists essentially perfoming 11 binary distinctions, each consisting in determining whether the term belongs or does not belong to any of the 11 fine-grained attitude types of Figure 3.1. Note that in Appraisal Theory a term can have more than one such attitude type. For example, the term `fair` is labeled in our lexicon with attitude types Quality, Propriety, and Veracity[1]. This means this is an *at-least-1*-of-n task, for $n = 11$, since the task is defined only on terms that carry appraisal, and which thus belong to at least one of the attitude type classes. Note also that the 11 attitude types are leaves in a hierarchy. This also allows us, if desired, to apply a hierarchical classification method, whereby the structure of the hierarchy is taken into account.

Thus, in determining attitude type we have considered two alternative classification methods:

Flat : this method simply ignores the fact that the categories are organized into a hierarchy and performs *multi-class* classification, assigning leaf categories from the taxonomy to each term; this gives a set of 11 different categories $C = \{c_1, \ldots, c_{11}\}$, corresponding to the 11 finest-grained attitude types. For each category i a binary classifier $\hat{\Phi}_i$ is generated by using all the terms in Tr_i^K as positive examples and all terms not belonging to Tr_i^K as negative examples.

Hierarchical : this method generates binary classifiers $\hat{\Phi}_j$ for each leaf *and* for each internal node. For an internal node c_j, as the set of positive training examples, the union of the sets of positive training examples of its descendant categories is used. For each node c_j (be it internal or leaf), as the set of negative examples, the union of the positive training examples of its sibling categories (minus possible positive training examples of c_j) is used. Both choices follow consolidated practice in the field of hierarchical categorization [28]. At classification time, test terms are classified by the binary classifiers at internal nodes, and only the ones that are classified as belonging to the node percolate down to the lower

[1] Out of a total of 1,847 terms in the test lexicon, 192 have more than one attitude type assigned.

levels of the tree. The hierarchical method has the potential advantage of using more specifically relevant negative examples for training, although the training sets for the lower-level categories will be smaller.

3.4.3 Force classification

Force is a simpler case, with four categories where each term belonging to exactly one of the four. Since the categories (Low, Median, High, and Max) are ordered along a scale of value, deciding which one applies to a given term is an *ordinal regression* problem. However, we (suboptimally) have dealt the problem as a 1-of-n *classification* problem (thereby disregarding the order among the categories), with n=4. We have deferred the use of ordinal regression for this problem to future work.

3.5 Experiments

We have examined the use of two learners for this task: (i) multinomial Naive Bayes, using Andrew McCallum's Bow implementation[2], and (ii) (linear kernel) Support Vector Machines, using Thorsten Joachims' SVMlight implementation [3].

We have also compared three possible classification modes for combining binary classifiers for a multiple labeling problem: (i) m-of-n, which may assign zero, one, or several classes to the same test term; (ii) *at-least-1* (of-n), a variant of m-of-n which always assigns one class when m-of-n would assign no class; (iii) 1-of-n, which always assigns exactly one class. Note that, from what we have said in Section 3.4, the a priori optimal approaches for classifying according to attitude type and force are (ii) and (iii), respectively. However, we have run experiments in which we have tested each of (i)-(iii) on both attitude and force. There are several justifications for this; for instance, trying (i) on attitude type is justified by the fact that forcing at least one category assignment, as *at-least-1*-of-n does, promises to bring about higher recall but lower precision, and nothing guarantees that the balance will be favorable. Suboptimal as some of these attempts may be, they are

[2] `http://www-2.cs.cmu.edu/~mccallum/bow/`

[3] `http://svmlight.joachims.org/`

legitimate provided that we use the correct evaluation measure for the task.

3.5.1 The base lexicon

The base lexicon we have used in the experiments is the one built by Whitelaw et al. [94]. It was constructed manually to give appraisal attribute values for a large number of evaluative adjectives and adverbs. Values for attitude type, orientation, and force are stored for each term. The lexicon also includes entries for modifiers, such as not and very, which are not used in our experiments. The lexicon was built starting from 400 terms and phrases extracted from examples for the different appraisal options in [65], then finding more candidate terms and phrases using WordNet and two online thesauri. Candidates were then manually checked and assigned attribute values, finally producing a lexicon of 1,847 terms.

The attitude type dimension of the corpus is defined by 11 different leaf categories, as described in Section 3.2, each one containing 189 terms on the average (the maximum is 284 for Affect, the minimum is 78 for Balance); every term is labeled by at least one and at most three categories (the average being 1.12). The hierarchy of the attitude taxonomy is displayed in Figure 3.1.

Force comprises four values in the corpus: Low (e.g., adequate), Median (e.g., good), High (e.g., awesome), and Max (e.g., best). Most (1464) entries in the corpus have Median force, with 30 Low, 323 High, and 57 Max.

Regarding the definition of the seed sets Tr_i^0 for the various categories c_i, as requested by our method, in the case of attitude and force we have faced the problem of selecting representative terms for all the categories of each task. We have decided to proceed by running *10-fold cross validation* experiments, i.e. splitting the lexicon in ten parts, then running ten experiments where nine tenth of the lexicon are used for training and the produced classifier is tested on the remaining one tenth. To guarantee that, for each of the ten experiments, each category c_i is adequately represented both in the training and in the test set, we have produced a split of the lexicon where in each tenth each category has (with the best possible approximation) a number of terms proportional to the global number of terms belonging to the category.

3.5.2 Evaluation measure

For evaluation, since multi-label nature of the classification tasks, where each category is rather unbalanced (i.e. on the average, there are many fewer terms belonging to a category than not belonging to it), we have used the well-known F_1 [62] measure.

This is defined as the harmonic mean of precision and recall:

$$\pi = \frac{TP}{TP+FP} \tag{3.1}$$

$$\rho = \frac{TP}{TP+FN} \tag{3.2}$$

$$F_1 = \frac{2\pi\rho}{\pi+\rho} = \frac{2TP}{2TP+FP+FN} \tag{3.3}$$

where, TP stands for true positives (the number of time any item has been correctly *assigned* to a category), FP for false positives (the number of time any item has been erroneously *assigned* to a category), and FN for false negatives (the number of time any item has been erroneously *not assigned* to a category). Note that F_1 is undefined when $TP + FP + FN = 0$, i.e. there are no positive examples. However, in our experiments there is always a positive example for every category, thus the F_1 measure is always defined.

We have computed both *microaveraged* F_1 (denoted by F_1^μ) and *macroaveraged* F_1 (F_1^M). F_1^μ is obtained by (i) computing the category-specific values TP_i, FP_i, and FN_i, (ii) obtaining TP as the sum of the TP_i's (same for FP and FN), and then (iii) applying Equation 3.3. F_1^M is obtained by (i) computing the category-specific precision and recall scores ($\pi_i = \frac{TP_i}{TP_i+FP_i}$ and $\rho_i = \frac{TP_i}{TP_i+FN_i}$), (ii) computing macroaveraged precision and macroaveraged recall (π^M and ρ^M) as the unweighted averages of the category-specific values π_i and ρ_i, respectively, and then (iii) applying Equation 3.3; this corresponds to taking the harmonic mean of π^M and ρ^M.

Using accuracy as the effectiveness measure when the categories are unbalanced has the consequence that the "trivial" classifier that always chooses the majority class turns out to be very effective. When the categories are *extremely* unbalanced the trivial classifier may turn out to be be more effective than any classifier generated through genuine effort. This does not happen when using F_1.

3.6 Results

We have ran evaluations for all combinations of learning algorithm (NB and SVM), classification model (flat and hierarchical), and classification method (n-of-m, at-least-1, and exactly-1); we have also considered the effect of using glosses from parts-of-speech other than the appropriate adjectives and adverbs, to see how stable our method is in the face of such ambiguity. Table 3.2 summarizes our results, comparing the effects of different values for each independent variable by averaging over results for the other variables.

3.6.1 Attitude type

The best results are clearly achieved by Naive Bayes; this result holds also for the non-averaged results of individual runs. Surprisingly, the flat classification model works noticeably better than the hierarchical model, which may indicate that the shared semantics of taxonomic siblings is not well-represented in the WordNet glosses.

Regarding classification methods, while the multilabeling n-of-m and at-least-1 methods achieve the highest precision and recall, respectively, the exactly-1 method achieves the best balance between the two, as measured by F_1. This may be explained by the relatively low average ambiguity of the lexicon (1.12 label per terms on the average). In practice, the higher recall method is probably preferred, since incorrect category assignments may be weeded out at the text analysis stage. Finally, we note that including glosses from parts-of-speech other than those in the lexicon did not appreciably change results.

3.6.2 Force

As for attitude type, Naive Bayes dominates for recall and F_1, while SVMs achieve better precision. Also similar is that at-least-1 classification increases recall at the expense of precision; exactly-1, which is a correct method for force (as it is unambiguous), achieves slightly better (macroaveraged) F_1 than m-of-n, but the difference is slight.

More significant, however, is that micro- and macroaveraged F_1 are quite different for force, showing that the majority category, Median, comprising 78% of terms, is better classified than other classes, though results still indicate that minority classes are being identified with rea-

Table 3.2: Summary of averaged cross-validation results, showing microaveraged (π^μ, ρ^μ, F_1^μ) and macroaveraged (π^M, ρ^M, F_1^M) statistics. Each row shows the average over all runs for given values for certain independent variables, averaging over all others (indicated by **–avg–**). The highest value in each column for each set of comparable results is boldfaced for ease of reading.

Dimension	Algorithm	Model	Method	POS	π^μ	ρ^μ	F_1^μ	π^M	ρ^M	F_1^M
attitude	NB	**–avg–**	**–avg–**	**–avg–**	**0.320**	**0.397**	**0.332**	0.362	**0.376**	**0.305**
attitude	SVM	**–avg–**	**–avg–**	**–avg–**	0.254	0.237	0.223	**0.464**	0.233	0.186
attitude	**–avg–**	flat	**–avg–**	**–avg–**	**0.381**	**0.421**	**0.371**	0.389	**0.401**	**0.345**
attitude	**–avg–**	hier	**–avg–**	**–avg–**	0.192	0.213	0.184	**0.437**	0.208	0.147
attitude	**–avg–**	**–avg–**	n-of-m	**–avg–**	**0.334**	0.222	0.237	**0.509**	0.225	0.207
attitude	**–avg–**	**–avg–**	at-least-1	**–avg–**	0.243	**0.375**	0.285	0.388	**0.357**	0.253
attitude	**–avg–**	**–avg–**	exactly-1	**–avg–**	0.284	0.353	**0.310**	0.343	0.331	**0.277**
attitude	**–avg–**	**–avg–**	**–avg–**	Adj,Adv	0.286	**0.318**	0.277	0.411	0.305	0.245
attitude	**–avg–**	**–avg–**	**–avg–**	Adj,Adv,V	0.285	**0.318**	0.277	0.412	**0.306**	0.246
attitude	**–avg–**	**–avg–**	**–avg–**	Adj,Adv,N	**0.289**	0.317	**0.279**	**0.417**	0.303	**0.247**
attitude	**–avg–**	**–avg–**	**–avg–**	Adj,Adv,V,N	0.287	0.315	0.277	0.413	0.303	0.245
force	NB	*n/a*	**–avg–**	**–avg–**	0.585	**0.732**	**0.634**	0.281	**0.614**	**0.352**
force	SVM	*n/a*	**–avg–**	**–avg–**	**0.586**	0.498	0.499	**0.662**	0.214	0.187
force	**–avg–**	*n/a*	n-of-m	**–avg–**	**0.755**	0.759	**0.757**	**0.501**	0.404	**0.305**
force	**–avg–**	*n/a*	at-least-1	**–avg–**	0.591	**0.806**	0.661	0.476	**0.487**	0.288
force	**–avg–**	*n/a*	exactly-1	**–avg–**	0.688	0.688	0.688	0.473	0.406	0.280
force	**–avg–**	*n/a*	**–avg–**	Adj,Adv	0.677	0.750	0.701	0.489	0.432	0.290
force	**–avg–**	*n/a*	**–avg–**	Adj,Adv,V	0.677	0.750	0.701	0.479	0.430	0.291
force	**–avg–**	*n/a*	**–avg–**	Adj,Adv,N	**0.680**	**0.753**	**0.704**	**0.490**	**0.434**	0.291
force	**–avg–**	*n/a*	**–avg–**	Adj,Adv,V,N	0.679	**0.753**	**0.704**	0.475	0.433	**0.292**

Table 3.3: 10-fold cross-validation results for term classification by attitude type using Naive Bayes with flat n-of-m categorization. Three different levels of expansion (K) of the training sets are reported ($K = 0$ means no expansion).

K	π^{μ}	ρ^{μ}	F_1^{μ}	π^{M}	ρ^{M}	F_1^{M}
0	.338	.484	**.398**	.306	.502	**.380**
1	.316	.478	.380	.293	.495	.368
2	.305	.467	.369	.287	.480	.359

sonable accuracy. Treatment of force in the future as an ordinal regression problem may help with this issue.

3.6.3 Expansion

Table 3.3 reports results for attitude type of applying expansion to the training sets, as described in Section 3.4.1. In contrast to previous results for orientation, expansion results in decreased effectiveness: the change in F_1^{μ} is -5.3% for $K = 1$ and -7.3% for $K = 2$. This is likely due to the fact that the use of synonymy and antonymy relation is a too coarse approach to the fine-grained nature of the taxonomy, and thus terms from different types of appraisal which are rather semantically close (e.g. Propriety with the term `corrupt` and Veracity with the term `honest`) mix up in the training sets, producing the observed degradation in results.

3.7 Conclusions

In this chapter we have shown how information contained in dictionary glosses can be exploited to automatically determine the types and forces of attitudes of terms. When put together with the similar methods for determining orientation and subjectivity, this method enables the automatic construction of lexicons in which a variety of sentiment-related attributes are attributed to words for use in appraisal extraction and OM [88, 94]. We have also found that, in contrast with previous work on orientation and subjectivity classification, lexical relation-based expansion of the base lexicon did not improve classification accuracy, probably because of the finer-grained nature of the classification task.

Chapter 4

SentiWordNet: Determining Opinion-Related Properties of Terms Senses

Abstract

In this chapter we face the problem of deteminig the subjectivity and orientation properties of the distinct *senses* of a term. The choice of using this finer grain in our analysis is motivated by the fact that terms may be ambiguous, and a term may have different subjectivity and orientation properties with respect to the specific sense they are intended to express.

We describe SentiWordNet, a lexical resource produced by asking an automated classifier $\hat{\Phi}$ to associate to the unique sense represented by each synset s of WordNet (version 2.0) a triplet of scores $\hat{\Phi}(s, p)$ (for $p \in P =\{\mathsf{Positive}, \mathsf{Negative}, \mathsf{Objective}\}$) describing how strongly that sense enjoy each of the three properties. The method used to develop SentiWordNet is based on the term classification method described in previous Chapters 1 and 2. The score triplet is derived by combining the results produced by a committee of eight ternary classifiers, all characterized by similar accuracy levels but extremely different classification behaviors.

4.1 Introduction

In Chapters 1, 2 and 3 we have focused on the problem of recognizing some opinion-related properties of *terms*, in the "related work" section of each chapter we have described some of the other relevant works which have focused of the same or similar tasks. An implicit hypothesis in all these works, including our, is that the label assigned to a term is *statistically relevant with respect to the various sense the term may have*, i.e., the label assigned to the term reflects the properties of its most frequent meaning. Another way to state this hypothesis is that the opinion-related properties are considered to be tolerant to the ambiguity of the term, or that the cases where this fails are irrelevant. For example the term `nice` is considered to be Positive, but the occurrences of the name of the French city of `Nice` is clearly Objective. In this case capitalization and the different POS can be used to distinguish the two cases (some works, e.g. [44, 53, 81], distinguish between different POSs of a word, and also our method can be easily modified to distinguish POSs). The case of the adjective `estimable` is a harder one, because POS and orthographic features cannot help in distinguishing between the Positive sense of "deserving respect" and the Objective sense of "which possible to be measured" (other terms exhibiting the same ambiguity are `short`, `straight`,`blue`, `ill`).

In this chapter we tackle a finer-grained problem, working on the determination the sentiment-related properties of *term senses*. Other works on this topic are those Andreevskaia and Bergler [3], Wiebe and Mihalcea [96], and Ide [48], which are described and compared with our in Section 4.2.

4.1.1 Our proposal

We propose SentiWordNet, a lexical resource produced by asking an automated classifier $\hat{\Phi}$ to associate to each synset[1] s of WordNet (version 2.0), a triplet of numerical scores $\hat{\Phi}(s,p)$ (for $p \in P =\{$Positive, Negative, Objective$\}$) describing how strongly the terms contained in s

[1] As already described in Chapter 1, Section 1.4.4, a WordNet synset represent a unique concept, which is defined by a unique gloss and is associated to a set of terms which can be used to represent that concept, and all with the same POS, each one associated to a sense number, (e.g., the adjectives `blasphemous(J,2)`, `blue(J,4)`, `profane(J,1)` are all contained in the same synset, whose sense is defined by the gloss "`characterized by profanity or cursing`").

enjoy each of the three properties. The assumption that underlies our switch from terms to synsets is that different senses of the same term may have different opinion-related properties.

Each of the three $\hat{\Phi}(s, p)$ scores ranges from 0.0 to 1.0, and their sum is 1.0 for each synset s. This means that a synset may have nonzero scores for all of the three categories, which would indicate that the corresponding terms have, in the sense indicated by the synset, each of the three opinion-related properties only to a certain degree.

For example, SentiWordNet attributes to the synset `[estimable(J,3)]`[2], corresponding to the sense "`may be computed or estimated`" of the adjective `estimable`, an Objective score of 1.0 (and Positive and Negative scores of 0.0), while it attributes to the synset `[estimable(J,1)]`, corresponding to the sense "`deserving of respect or high regard`", a Positive score of 0.75, a Negative score of 0.0, and an Objective score of 0.25 (see Figure 4.2).

Note that associating a graded score to a synset for a certain property (e.g., Positive) may have many different interpretations, for example:

1. the terms in the synset are Positive only to a certain degree, i.e. with a certain *intensity*;
2. the terms in the synset are sometimes used in a Positive sense and sometimes not, e.g., depending on the context of use;
3. the annotator is uncertain whether the terms in the synset are Positive.

Interpretation 1 has a *fuzzy* character, implying that each instance of these terms, in each context of use, has the property to a certain degree, and that the annotator is certain of this degree.

Interpretation 2 has a *probabilistic* nature (of a frequentist, "objective" type), implying that membership of a synset in the set denoted by the property must be computed by counting the number of contexts of use in which the terms have the property.

[2] We here adopt the convention according to which a term enclosed in square brackets denotes a synset; thus [`poor(J,7)`] refers not to the term `poor` but to the synset consisting of *adjectives* {`inadequate(J,2)`, `poor(J,7)`, `short(J,4)`}. The sense numbers associated to each term refer to the WordNet 2.0 sense numbers. The other POS used in WordNet are nouns (N), verbs (V), and adverbs (R).

Interpretation 3 has, again, a *probabilistic* nature, but of a "subjective" type, i.e. related to the degree of confidence that the annotator has in the membership of the synset in the set denoted by the property.

The above presented interpretations are just three reasonable models of the semantic of the assignment of graded scores to synsets, other can be derived by combination of them, and many other arise directly by the methods proposed to automatically assign the scores. Related to terms, rather than synsets, a similar intuition of graded scoring have previously been presented in [54], whereby a term could have both a Positive and a Negative orientation, each to a certain degree. Non-binary scores are attached to opinion-related properties also in [92] (see Chapter 1, Section 1.2.2; the authors' interpretation of these scores is related to the confidence in the correctness of the labeling, rather than in how strongly the term is deemed to possess the property. A related point has recently been made in [2], in which terms that possess a given opinion-related property to a higher degree are claimed to be also the ones on which human annotators asked to assign this property agree more.

However, most of these models of interpretation, although inherently different, are closely related one to the others that, in fact, are not distinguished in practical applications. We discuss the interpretation model of SentiWordNet scores in Section 4.3.4.

We believe that a graded (as opposed to binary) evaluation of the opinion-related properties of terms can be helpful in the development of opinion mining applications. A binary classification method will probably label as Objective any term that has no strong subjectivity, e.g., terms such as `short` or `alone`. If a sentence contains many such terms, a resource based on a binary classification will probably miss its subtly subjective character, while a graded lexical resource like SentiWordNet may provide enough information to capture such nuances.

The method we have used to develop SentiWordNet is based on the term classification method described in Chapters 1 and 2. In the case of SentiWordNet the method relies on the quantitative analysis of the *glosses* associated to each single synsets, and on the use of the resulting vectorial term representations for semi-supervised synset classification. The triplet of scores is derived by combining the results produced by a committee of eight ternary classifiers, each of which has demonstrated, in our previous tests, similar accuracy but different characteristics in terms of classification behavior. Two versions of SentiWordNet are discussed and evaluated in this chapter, which are obtained by two different methods of generating the eight classifiers and combining their results.

4.1.2 Chapter outline

The rest of the chapter is organized as follows. Section 4.2 discusses related work. Section 4.3 describes the semi-supervised learning method by which SentiWordNet has been built, describing how the classifiers have been trained (Section 4.3.1) and combined (Section 4.3.2). Sections 4.4 and 4.5 describe the evaluation environment and the results of an evaluation exercise by which we have attempted to estimate the accuracy of SentiWordNet. Section 4.6 concludes.

4.2 Related work

4.2.1 Andreevskaia and Bergler

One work that deals with tagging synsets by subjectivity and orientation is the recent work from Andreevskaia and Bergler [3], which is based on the crude idea of tagging with a category $p \in \{\mathsf{Positive}, \mathsf{Negative}, \mathsf{Objective}\}$ all synsets whose eXtendedWordNet gloss[3] contains (i) a synset that is known to belong to p, or (ii) a synset that is reachable from synsets belonging to p via WordNet lexical relations that (similarly to what we do in Section 4.3.1) can be assumed to preserve opinion-related properties.

However, there are key differences between [3] and our work. First, those authors limit their work to WordNet adjectives, while we tag all WordNet synsets, irrespectively of their POS; arguably, words other than adjectives are the hardest to work with, since they tend to be sentiment-laden to a much smaller degree than adjectives (see Table 4.7). Second, the system of [3] tags synsets as either belonging or not belonging to a category p, while in our system membership is a matter of degrees. Last, [3] have not evaluated the accuracy of their system at tagging *synsets* (they have only indirectly evaluated their system by tagging the General Inquirer, which is a set of manually tagged *terms*,

[3] eXtendedWordNet [43] is a version of WordNet in which, among other things, all terms appearing in the gloss of a synset are (automatically) disambiguated, and are thus linked to the synset they pertain to (see also Chapter 5, Section 5.5.1).

since at the time of their work no gold standard of manually tagged synsets existed.

4.2.2 Wiebe and Mihalcea

Wiebe and Mihalcea [96] have proposed a method for assigning subjectivity scores to term senses. The method is based of finding distributionally similar *terms* to the observed *term sense* and then analyzing their occurrences in an annotated corpus (the MPQA Opinion Corpus [97], see also Chapter 6, Section 6.3.2), where expression of subjectivity are annotated. The more the terms appear in subjective expressions the more the term sense gets a high subjectivity score. The terms identified to be highly related to the term sense contribute more to its subjectivity score than the one less related.

A relevant difference from their method to our is that it requires a large manually annotated text corpus to have a statistically significant database on which to compute the subjectivity score, while our method requires just three small seed sets of synsets. The size of the annotated corpus not only affects the quality of results but also the actual possibility to produce some results: in the evaluation the authors have evaluated their method on a set of 354 senses from 64 words, but only 272 have been actually evaluated, because for the remaining 82 sense none of their distributionally similar terms occur in the MPQA Corpus.

They have used the subjectivity information on sense to perform a word-sense disambiguation task (WSD), showing a significant 2.2% reduction in *error*.

4.2.3 Ide

Another work facing the problem of tagging senses of terms is the one of Ide [48]. This work propose a method for assigning the proper WordNet senses to a set of terms conceptually related. The method takes in input a set of terms representing a concept, and identifies which WordNet senses for each term are the ones which determine the inclusion of the term in the list. The core idea of the method is to apply mutual disambiguation between pair of terms in the list to identify the most related the senses of the two terms, which by hypothesis are those which determine the inclusion of both terms in the list. The method has been used to assign WordNet senses to the lexical units of the categories of

Table 4.1: The a-labels from WordNet-Affect, with example terms.

A-Labels	Examples
EMOTION	anger(N,1), fear(V,1)
MOOD	animosity(N,1), amiable(J,1)
TRAIT	aggressiveness(N,1), competitive(J,1)
COGNITIVE STATE	confusion(N,2), dazed(J,2)
PHYSICAL STATE	illness(N,1), all_in(J,1)
EDONIC SIGNAL	hurt(N,3), suffering(N,4)
EMOT.-ELICITING SIT.	awkwardness(N,3), endangered(J,1)
EMOTIONAL RESPONSE	cold sweat(N,1), tremble(V,2)
BEHAVIOUR	offense(N,1), inhibited(J,1)
ATTITUDE	intolerance(N,1), defensive(N,1)
SENSATION	coldness(N,1), feel(V,3)

FrameNet [6] and also to terms in the General Inquirer lexicon. There are two fundamental differences between this method and our: (i) it requires the complete set of terms to be already classified as belonging to the category under examination, while our starts from a small set of examples; (ii) it performs a hard classification of senses, opposite to our graded scoring.

4.2.4 Valitutti et al.

Valitutti et al. [93] have developed WordNet-Affect, which identifies as set of 1,314 WordNet synsets and 3,340 terms that are related to affective concepts. Affective concepts are classified into a taxonomy of 11 categories (*a-label*). Table 4.1 show the possible a-labels with examples. The resource have been manually developed starting from a set of 1,903 terms selected from various sources which have then linked to their related WordNet synset, each one with an associated *frame* of related information (e.g. Italian and English version, a-label). This work is not closely related to our but is indeed another interesting resource related to subjectivity expression.

4.3 Building SentiWordNet

The method we have used to develop SentiWordNet relies on automatically training eight individual *synset classifiers* $\hat{\Phi}_1(s,p), \ldots, \hat{\Phi}_8(s,p)$, and then gathering them into a *(synset) classifier committee* $\hat{\Phi}(s,p)$.

Synset classifiers (be them individual classifiers or classifier committees) are ternary classifiers, i.e., they attempt to predict whether a synset is Positive, Negative, or Objective. By an *n-ary classifier* we here mean a function $\hat{\Phi} : S \times P \rightarrow [0,1]$ that, given an object s and a class $p \in P = \{p_1, \ldots, p_n\}$, returns a numerical score $\hat{\Phi}(s,p)$.

Scores can be binary-valued or real-valued. In the former case, $\hat{\Phi}(s,p)$ must equal 1 for a single $p_i \in P$ and 0 for all $p \in P/\{p_i\}$; this corresponds to deciding that s belongs to class p_i and does not belong to any class in $P/\{p_i\}$.

In the latter case $\hat{\Phi}(s,p)$ denotes the confidence, or degree of belief, that the classifier has in the fact that s has indeed property p (the higher the value, the higher the confidence). If a binary decision needs to be taken, synset s is deemed to belong to the class

$$\arg\max_p \hat{\Phi}(s,p)$$

that has received the highest score.

Section 4.3.1 will deal with the method we have used for training the $\hat{\Phi}_i$'s, while Section 4.3.2 will discuss the issue of building a classifier committee $\hat{\Phi}$ out of them.

4.3.1 Training synset classifiers

The method we have used to develop the individual classifiers $\hat{\Phi}_1, ..., \hat{\Phi}_8$ is an adaptation to synset classification of our term classification method used to classify term by subjectivity and orientation (see Chapter 2). The *seed sets* to be fed to the algorithm in this case will contain synsets instead of terms. We have defined the two seed sets for the Positive and Negative categories, Tr_p^0 and Tr_n^0, by manually selecting the intended synsets for the 14 “paradigmatic” terms (e.g., the Positive term `nice`, the Negative term `nasty`) which were used as seed terms by Turney [92]. For example, for the term `nice` we have removed the synset relative to the French city of Nice. The process has resulted in 47 Positive and 58 Negative synsets. The seed sets are then iteratively expanded, and every expansion step consists in:

1. adding to Tr_p^k (resp. Tr_n^k) all the synsets that are connected to synsets in Tr_p^{k-1} (resp. Tr_n^{k-1}) by WordNet lexical relations (e.g., *also-see*) such that the two related synsets can be assumed to have *the same* orientation[4];

2. in adding to Tr_p^k (resp. Tr_n^k) all the synsets that are connected to synsets in Tr_n^{k-1} (resp. Tr_p^{k-1}) by WordNet lexical relations (e.g., *direct antonymy*) such that the two related synsets can be assumed to have *opposite* PN-polarity.

The relations we have used in tasks working on terms (see Chapter 1, Section 1.4.3 and Chapter 2, Section 2.4.2) are synonymy (for use in substep 1) and direct antonymy (for use in substep 2) between terms, as also is common in related literature [53, 54, 89]. In the case of synsets, synonymy cannot be used because it is the relation that defines synsets, thus it does not connect different synsets. We have then followed the method used in [93] for the development of WordNet-Affect (see Section 4.2.4): after hand-collecting a number of labeled terms from other resources, Valitutti and colleagues generate WordNet-Affect by adding to them the synsets reachable by navigating the relations of *direct antonymy*, *similarity*, *derived-from*, *pertains-to*, *attribute*, and *also-see*, which they consider to reliably preserve/invert the involved labels. Given the similarity with our task, we have used exactly these relations in our expansion. The final sets Tr_p^K and Tr_n^K, along with the set Tr_o^K described below, are used to train the ternary classifiers.

The Tr_o^0 seed set is treated differently from Tr_p^0 and Tr_n^0, because of the inherently "complementary" nature of the Objective category (an Objective term can be defined as a term that does *not* have either Positive or Negative characteristics). We have heuristically defined Tr_o^0 as the set of synsets that (a) do not belong to either Tr_p^K or Tr_n^K, and (b) contain terms not marked as either Positive or Negative in the General Inquirer lexicon [86]; this lexicon was chosen since it is, to our knowledge, one of the largest manually annotated lexicon in which terms are tagged according to the Positive or Negative categories. The resulting Tr_o^0 set consists of 17,530 synsets; for any K, we define Tr_o^K to coincide with Tr_o^0. As usually done for terms, each synset is then given a vectorial representation, obtained by applying a standard text indexing

[4] All the synsets that are in Tr_p^{k-1} (resp. Tr_n^{k-1}) are added to Tr_p^k (resp. Tr_n^k) by default.

technique (cosine-normalized $tf * idf$ preceded by stop word removal) to its gloss, which we thus take to be a textual representation of its semantics. In Section 4.3.2 we discuss two different methods (called Combination Method A or Combination Method B) by which we combine the $\hat{\Phi}_1(s,p), \ldots, \hat{\Phi}_8(s,p)$ into a committee $\hat{\Phi}(s,p)$. Combination Method A requires the $\hat{\Phi}_i$'s to output binary scores, while Combination Method B requires them to output real-valued scores. As a consequence, we use the vectorial representations in two different ways[5], dubbed Learning Method A and Learning Method B, according to whether Combination Method A or Combination Method B are going to be used:

1. In Learning Method A, the $\hat{\Phi}_i$'s are obtained by means of supervised learners that generate binary classifiers. The vectorial representations of the training synsets are input to a supervised learner which generates two binary classifiers $\hat{\Phi}_i^p$ and $\hat{\Phi}_i^n$: $\hat{\Phi}_i^p$ must discriminate between terms that belong to the Positive category and ones that belong to its complement (**not** Positive), while $\hat{\Phi}_i^n$ must discriminate between terms that belong to the Negative category and ones that belong to its complement (**not** Negative).

   In the training phase, the terms in $Tr_n^K \cup Tr_o^K$ are used as training examples of category (**not** Positive), and the terms in $Tr_p^K \cup Tr_o^K$ are used as training examples of category (**not** Negative).

2. Terms that have been classified *both* into Positive by the $\hat{\Phi}_i^p$ and into (**not** Negative) by $\hat{\Phi}_i^n$ are deemed to be positive, and terms that have been classified *both* into (**not** Positive) by $\hat{\Phi}_i^p$ and into Negative by $\hat{\Phi}_i^n$ are deemed to be negative. The terms that have been classified (i) into both (**not** Positive) and (**not** Negative), or (ii) into both Positive and Negative, are taken to be Objective.

   The two binary classifiers $\hat{\Phi}_i^p$ and $\hat{\Phi}_i^n$ working together thus implement, as is often the case in the supervised learning literature, a ternary classifier $\hat{\Phi}_i$ which returns a triplet of binary scores for the $p \in P$. This is then applied to the vectorial representations of all WordNet synsets s (including those in $Tr^K - Tr^0$).

3. If Learning Method B, the $\hat{\Phi}_i$'s are obtained by means of supervised learners that directly generate n-ary classifiers, since the

[5] These two different ways were called Approach II and Approach III in Chapter 2, Section 2.4.3.

resulting classifiers return a triplet of real-valued scores for the $p \in P$. In the training phase, the terms in Tr_p^K, Tr_n^K, and Tr_o^K are used as positive examples of Positive, Negative, and Objective, respectively.

The main difference between Learning Methods A and B is that in Learning Method B Objective is seen as a category, or concept, in its own right, while in Learning Method A objectivity is viewed as an unmarked category, i.e. as the "absence of subjectivity" (in fact, in Learning Method A the training examples of Objective are only used as training examples of the *complements* of Positive and Negative).

Note also that, while for Learning Method A we use well-known learners for binary classification (support vector machines using linear kernels, and the Rocchio learner), for Learning Method A we use their n-ary versions[6].

Note that other out of the three approaches we have presented in Chapter 2 we have chosen Approach II since it is the one that, in the experiments on terms, yielded the best effectiveness, and Approach III since for Combination Method B we needed the $\hat{\Phi}_i$ to output non-binary scores for each $p \in P$.

4.3.2 Defining the committee of classifiers

In Chapter 2, Section 2.5, we point out how different combinations of training set and learner behave in a radically different way, even though with similar levels of accuracy. The main three observations we recall here are the following:

- Low values of K produce small training sets for Positive and Negative, which produces binary classifiers with low recall and high precision for these categories. By increasing K these sets get larger, and the effect of these larger numbers is to increase recall but to also add "noise" to the training set, which decreases precision.

[6] The Rocchio learner we have used is from Andrew McCallum's *Bow* package (`http://www-2.cs.cmu.edu/~mccallum/bow/`), while the SVMs learner we have used is Thorsten Joachims' SVM^{light} (`http://svmlight.joachims.org/`), version 6.01. Both packages allow the respective learners to be run in n-ary (aka "multiclass") fashion.

- Learners that use information about the prior probabilities of categories, which estimate these probabilities from the training sets, are sensitive to the relative cardinalities of the training sets, and tend to classify more items into the categories that have more positive training items. Learners that do not use this kind of information, like Rocchio, do not exhibit this kind of behaviour.

- The difference in behaviour mentioned above does not affect the overall accuracy of the method, but only the relative proportions of items classified as Positive ∪ Negative and items classified as Objective, while the accuracy in discriminating between Positive and Negative items tends to be constant.

It is a well-known fact of computational learning theory that, the more independent from each other a set of classifiers are, the better they perform once assembled into a committee [90]. Since the above-mentioned difference in behaviour among our classifiers is a witness of their independence, we have decided to combine different configurations of training set and learner into a committee.

Specifically, we have defined four different training sets, by choosing four different values of $K \in \{0, 2, 4, 6\}$, and we have alternatively used two learners (Rocchio and SVMs); this yields a total of eight ternary classifiers. With $K = 0$, SVMs produced very "conservative" binary classifiers for Positive and Negative, i.e. classifiers characterized by very low recall and high precision. For $K = 6$, SVMs produced instead very "liberal" binary classifiers for Positive and Negative, i.e. classifiers that tend to classify many synsets as Positive or Negative even in the presence of very little evidence of subjectivity. The Rocchio learner has a similar behaviour, although not dependent on the prior probabilities of categories. As mentioned above, we have experimented with two different combination methods for computing the final triplets of $\hat{\Phi}(s, p)$ scores:

- In Combination Method A, we use ternary classifiers $\hat{\Phi}_i$ generated by Learning Method A, which thus return a triplet of binary scores; $\hat{\Phi}_i$ assigns to s exactly one of the three classes in P (i.e. $\hat{\Phi}_i(s, p) = 1$ for one $p \in P$ and $\hat{\Phi}_i(s, p) = 0$ for the other two $p \in P$). The final scores $\hat{\Phi}(s, p)$ are determined by the (normalized) proportion of ternary classifiers that have assigned the corresponding label to s, i.e.,

$$\hat{\Phi}(s,p) = \frac{1}{8}\sum_{i=1}^{8} [\![\hat{\Phi}_i(s) = p]\!] \tag{4.1}$$

where $[\![\pi]\!]$ indicates the characteristic function of predicate π (i.e. the function that returns 1 if π is true and 0 otherwise). If all the $\hat{\Phi}_i$'s agree in assigning the same label to a synset s, that label will have a score of 1.0 for s, otherwise each label will have a score proportional to the number of classifiers that have assigned it.

- In Combination Method B, we use ternary classifiers $\hat{\Phi}_i$ generated by Learning Method A, which thus return a triplet of real-valued scores; each ternary classifier $\hat{\Phi}_i$ first attaches three non-binary scores $\hat{\Phi}_i(s,p)$, for all $p \in P$, to each synset s. The final scores $\hat{\Phi}(s,p)$ are obtained by simply adding the corresponding real-valued scores from the $\hat{\Phi}_i$'s and then normalizing them, i.e.,

$$\hat{\Phi}(s,p) = \frac{\displaystyle\sum_{i=1}^{8} \hat{\Phi}_i(s,p)}{\displaystyle\sum_{p\in P}\sum_{i=1}^{8} \hat{\Phi}_i(s,p)} \tag{4.2}$$

Note that Combination Method B is “finer-grained” than Combination Method A, since the scores produced by Equation 4.1 range on the discrete set $\{0, \frac{1}{8}, \ldots, \frac{7}{8}, 1\}$, while the scores produced by Equation 4.2 range on the full real-valued [0,1] interval. Note also that, while Combination Method A only brings to bear the binary decisions of the individual classifiers $\hat{\Phi}_i$, Combination Method B also brings to bear the real-valued scores $\hat{\Phi}_i(s,p)$ that these classifiers have produced, i.e., the degrees of confidence that the $\hat{\Phi}_i$'s have in the correctness of their binary decisions. All in all, Combination Method B seems *a priori* conceptually more interesting than Combination Method A; we will experimentally evaluate them in Section 4.5.

Hereafter, by SentiWordNet 1.0 (resp. SentiWordNet 1.1) we will denote the result of classifying WordNet according to Learning and Combination Methods A (resp. B)[7].

[7] This naming convention is due to the fact that the first version of SentiWordNet we have publicly released was the one based on Learning and Combination

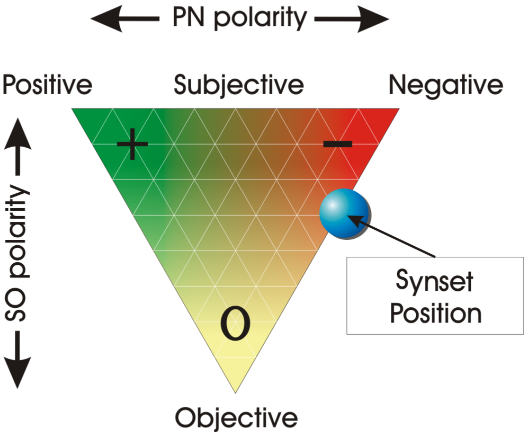


Figure 4.1: The graphical representation adopted by SentiWordNet for representing the opinion-related properties of a synset.

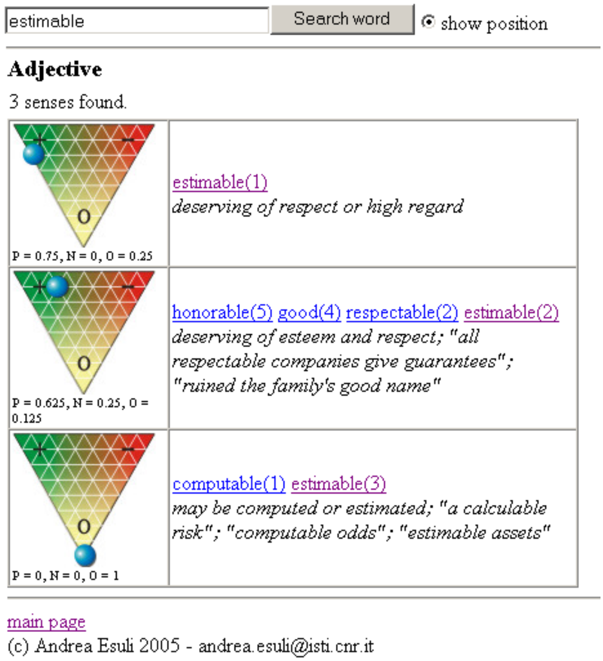
4.3.3 Visualizing SentiWordNet

Given that the sum of scores in a triplet is always 1.0, it is possible to display this triplet in a triangle whose vertices correspond to the maximum possible values for the three dimensions observed. Figure 4.1 shows the graphical model we have designed to display the score triplet associated to a synset. This visualization model is intended just to be a simple graphical representation of the scores and not an interpretation model of them, which is discussed in Section 4.3.4. This visualization model is adopted in the Web-based graphical user interface through which SentiWordNet can be accessed at `http://swn.isti.cnr.it/`. Figures 4.2 and 4.3 show two screenshots of the output for the synsets that include the terms `estimable` and `short`.

4.3.4 The interpretation model

How the scores assigned by SentiWordNet to synsets have to interpreted? As we argue in Section 4.1.1, there are many possible models of interpretation. For human-made resources the choice of the interpretation model is defined by the annotators who actually build the resource. For those produced by automatic methods, like our, the interpretation

Methods A



estimable [Search word] ● show position

Adjective

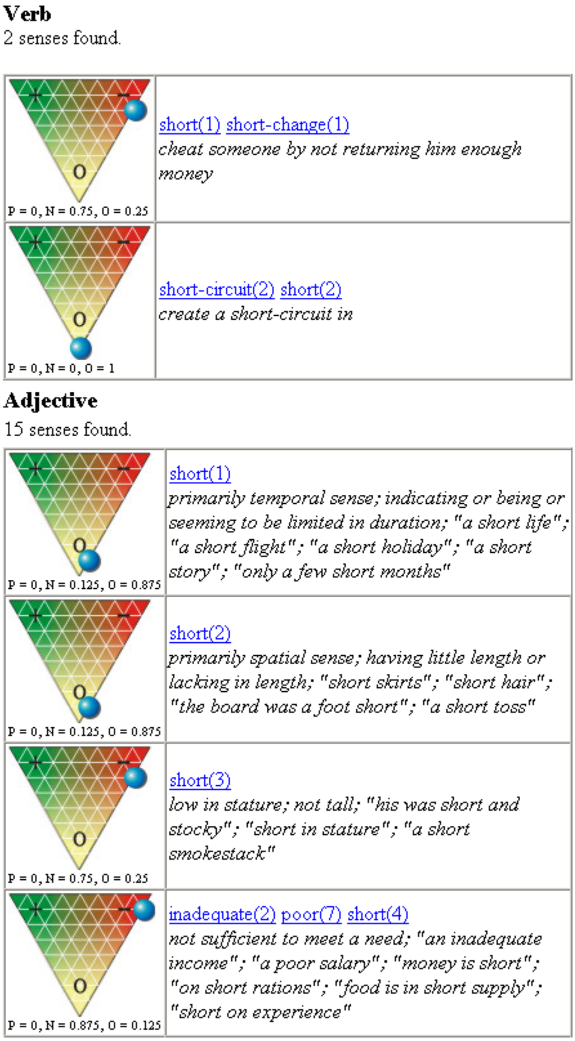
3 senses found.

P = 0.75, N = 0, O = 0.25	estimable(1) *deserving of respect or high regard*
P = 0.625, N = 0.25, O = 0.125	honorable(5) good(4) respectable(2) estimable(2) *deserving of esteem and respect; "all respectable companies give guarantees"; "ruined the family's good name"*
P = 0, N = 0, O = 1	computable(1) estimable(3) *may be computed or estimated; "a calculable risk"; "computable odds"; "estimable assets"*

main page

(c) Andrea Esuli 2005 - andrea.esuli@isti.cnr.it

Figure 4.2: SentiWordNet visualization of the opinion-related properties of the synsets that include the term `estimable` (actual scores are from SentiWordNet 1.0).



Verb
2 senses found.

P = 0, N = 0.75, O = 0.25	short(1) short-change(1) *cheat someone by not returning him enough money*
P = 0, N = 0, O = 1	short-circuit(2) short(2) *create a short-circuit in*

Adjective
15 senses found.

P = 0, N = 0.125, O = 0.875	short(1) *primarily temporal sense; indicating or being or seeming to be limited in duration; "a short life"; "a short flight"; "a short holiday"; "a short story"; "only a few short months"*
P = 0, N = 0.125, O = 0.875	short(2) *primarily spatial sense; having little length or lacking in length; "short skirts"; "short hair"; "the board was a foot short"; "a short toss"*
P = 0, N = 0.75, O = 0.25	short(3) *low in stature; not tall; "his was short and stocky"; "short in stature"; "a short smokestack"*
P = 0, N = 0.875, O = 0.125	inadequate(2) poor(7) short(4) *not sufficient to meet a need; "an inadequate income"; "a poor salary"; "money is short"; "on short rations"; "food is in short supply"; "short on experience"*

Figure 4.3: SentiWordNet visualization of the opinion-related properties of the synsets that include the term `short` (actual scores are from SentiWordNet 1.0).

model have to be derived by how the method works and which information uses.

For example, in [92] the orientation score (ranging in $[-1, 1]$) assigned to a term is related to the fact that the terms tends to co-occur more or less with a set of Positive *paradigmatic terms* with respect to a set of Negative ones. Thus, the assigned score can be interpreted as a *similarity of use* measure respect to well-known Positive and Negative terms.

Our method combines two interpretation models, one deriving by the classification method used, and the other deriving by the use of a committee of classifiers.

Each classifier $\hat{\Phi}_i(s, p)$ uses the information contained in a synset's gloss to classify it. The classification model is learned from a training set of glosses of *paradigmatic* synsets for the categories $p \in P$. Thus, we can say that a single classifier $\hat{\Phi}_i(s, p)$ uses a *similarity of description* model[8].

The committee classifier $\hat{\Phi}(s, p)$ combines the various $\hat{\Phi}_i(s, p)$ by averaging their results. As described in Section 4.3.2, the classifiers $\hat{\Phi}_i(s, p)$ in the committee are built using different parameters, so that some behave in a more "conservative" or "liberal" way than others in recognizing subjectivity. However each classifier in the committee is based on the same interpretation model described above. The averaging performed by the committee classifier can be interpreted, as reported in Point 3 of Section 4.1.1, as a *confidence of classification* measure. For example, in Combination Method A, if a synset s is classified as Positive by all the classifiers $\hat{\Phi}_i(s, p)$, the score assigned by $\hat{\Phi}(s, p)$ will be fully Positive; if a "conservative" classifier classifies s as Objective, the score assigned by $\hat{\Phi}(s, p)$ will be mainly Positive and slightly Objective, indicating a high confidence that s is Positive but also the possibility for s to be Objective[9].

4.4 Evaluating SentiWordNet

How reliable are the opinion-related scores attached to synsets in SentiWordNet? Fully testing the accuracy of our tagging method experimentally is impossible, since for this would require a version of Word-

[8] This applies to both Learning Methods A and B (see Section 4.3.1), because the data representation and text classification methods are the same.

[9] The same consideration applies to Combination Method B, which differs only for its finer grain.

Net manually annotated according to our three properties of interest, and the unavailability of such a manually annotated resource is exactly the reason why we are interested in generating it automatically.

A first, approximate indication of the quality of SentiWordNet can be gleaned by looking at the accuracy obtained by our method in classifying the General Inquirer [86] terms by subjectivity and orientation (see Chapter 2, Section 2.5). The reader should however bear in mind a few differences between the method used there and the one used here: (i) we here classify entire synsets, instead of terms, which can sometimes be ambiguous, and can thus be more difficult to classify correctly; (ii) as discussed in Section 4.3.1, the WordNet lexical relations used for the expansion of the training set are different. The effectiveness results reported on GI terms may thus be considered only approximately indicative of the accuracy of the SentiWordNet labels.

4.4.1 The Micro-WNOp gold standard

A second, more direct route to evaluating SentiWordNet is by using a manually annotated *subset* of WordNet as a "gold standard" against which to evaluate the scores attached to the same synsets in SentiWordNet. A subset of this kind, called Micro-WNOp, indeed exists [15][10]: it consists of 1,105 synsets manually annotated by a group of five human annotators (hereafter called J1, ..., J5); each synset is assigned a score for each of the three categories Positive, Negative, and Objective, with the scores in the triplet summing up to 1 for each synset. The authors reported that their intended interpretation model for assigning scores to synsets is mainly based on the criteria 1 described in Section 4.1.1.

Synsets 1-110 (here dubbed Micro-WNOp(1)) have been tagged by all the annotators working together, so as to develop a common understanding of the semantics of the three categories; then, J1, J2 and J3 independently tagged each synsets 111–606 (Micro-WNOp(2), while J4 and J5 independently tagged synsets 607–1105 (Micro-WNOp(3)).

It is also noteworthy that Micro-WNOp as a whole, and each of its subsets, are representative of the distribution of parts of speech in WordNet: this means that, e.g., if $x\%$ of WordNet synsets are nouns, also $x\%$ of Micro-WNOp synsets are nouns. Moreover, this property also

[10] Publicly available for download at: `http://www.unipv.it/wnop/`

holds for each single part (i.e. Micro-WNOp(x)) of Micro-WNOp.

The Web-based graphical user interface that was used by the annotators is based on the same graphical model as discussed in Section 4.3.3. In this interface each annotator was presented with a synset and was asked to place a bullet within the triangle in the position that represented, according to the him/her, the mix of the three opinion-related properties as possessed by the synset.

See [15] for further details on how Micro-WNOp and its subsets were designed.

The fact that the same synset has been tagged by more than one human annotator also allows measuring the rate of inter-annotator agreement, and relating it to the level of difficulty of the automated annotation task (see Section 4.5.

Note that 1,105 synsets correspond to less than 1% of the total 115,424 WordNet synsets; this clarifies that, again, the accuracy obtained on this gold standard may be considered only as indicative of the (unknown) level of accuracy with which SentiWordNet has been produced. Notwithstanding this, Micro-WNOp will prove a useful tool in the comparative evaluation of future systems that, like ours, tag WordNet synsets by opinion, including possible future releases of SentiWordNet.

4.4.2 Evaluation measure

To evaluate SentiWordNet on Micro-WNOp we have faced the problem of having triplets of numerical scores to compare. For example, what is the error made by SentiWordNet if it assigns to a synset s the triplet $\hat{\Phi}(s) = \{0.1, 0.0, 0.9\}$[11] when Micro-WNOp assigns to it the values $\Phi(s) = \{0.5, 0.0, 0.5\}$?

To solve this problem we have first simplified the scoring model to one where both Φ and $\hat{\Phi}$ may have values only in $\{0, 1\}$. In this simplified model only a category could have a score equal to one while the other two have a zero score, to respect the constraint that the sum of scores must be equal to one. The simplified model is thus equivalent to a *single-label* classification model, where the label $p \in P$ assigned to the synset s is the only one for which $\Phi(s, p) = 1$.

[11] In the following we will use this compact notation to list the Positive, Negative and Objective scores of a synset, in this order.

In the simplified model we can measure the **mean squared error** in classification with the formula:

$$MSE_{\mu} = \frac{\sum\limits_{p \in P} \sum\limits_{s \in G} \Phi(s,p)(\hat{\Phi}(s,p) - \Phi(s,p))^2}{\sum\limits_{p \in P} \sum\limits_{s \in G} \Phi(s,p)} \tag{4.3}$$

Considering the simplified model as a limit case of the one we actually have, with Φ and $\hat{\Phi}$ having values in the range $[0,1]$, we can see that Equation 4.3 still holds in this latter case. Given that foreach synsets s we have $\sum\limits_{p \in P} \Phi(s,p) = 1$ the denominator of Equation 4.3 still counts the total number of synsets in the gold standard. For the numerator the mean squared error evaluation is subdivided on the three categories in proportion to the score assigned to each by the gold standard.

For example, given a synset s with scores $\Phi(s) = \{0.0, 0.5, 0.5\}$ in the gold standard, we obtain $MSE_{\mu} = 1$ on it if the predicted value is $\hat{\Phi}(s) = \{1.0, 0.0, 0.0\}$, and $MSE_{\mu} = 0.125$ if the predicted value is $\hat{\Phi}(s) = \{0.5, 0.5, 0.0\}$.

Equation 4.3 evaluates a *micro-averaged* mean squared error measure, in fact it considers each synset to have equal relevance whatever its values in the gold standard are. This can result in an evaluation biased toward the quality of score assignment to synsets with a high Objective score, given that these synsets are the majority in the gold standard. To present an evaluation which gives equal weight to the three categories, we have also used a *macro-averaged* version of Equation 4.3:

$$MSE_p = \frac{\sum\limits_{s \in G} \Phi(s,p)(\hat{\Phi}(s,p) - \Phi(s,p))^2}{\sum\limits_{s \in G} \Phi(s,p)} \tag{4.4}$$

$$MSE_M = \frac{\sum\limits_{p \in P} MSE_p}{\#(P)} \tag{4.5}$$

where $\#(P)$ is the number of elements in P, three in our case. In the macro-averaged model the contribute of errors on each category is computed separately (MSE_p) and then averaged (MSE_M).

4.5 Results

We have used the two MSE_{μ} and MSE_{M} measures to compare SentiWordNet 1.0 and SentiWordNet 1.1 on the Micro-WNOp gold standard. To have a baseline reference we have defined three $\hat{\Phi}_b$ functions, one for each category, that return $\hat{\Phi}_b(s,p) = 1$ only when $p = b$. We have also made a pairwise evaluation of inter-annotator agreement study between the annotators J1, J2 and J3 on Micro-WNOp(2), and J4 and J5 on Micro-WNOp(3), using these results as reference values to estimate the absolute quality of the SentiWordNet data. Tables 4.2 and 4.3 shows the results of evaluating SentiWordNet 1.0 and 1.1, and the three baseline functions $\hat{\Phi}_b$, on the whole Micro-WNOp gold standard, and on each of its section separately. For the Micro-WNOp(2) and Micro-WNOp(3) sections we have derived the gold standard triplets by averaging the triplet scores of all the annotators of each section.

The results for the three baseline functions show that the $\hat{\Phi}_{Objective}$ function obtains the best values (lower is better), especially in terms of MSE_{μ}, as predicted in previous section. Both SentiWordNet 1.0 and 1.1 show relevant improvements respect to the $\hat{\Phi}_{Objective}$ baseline. The relative improvement of SentiWordNet 1.0 is more than 41% in terms of both MSE_{M} and MSE_{μ} measures. SentiWordNet 1.1 improvement is lower, but still over 22%. These figures are confirmed in the section-by-section analysis of Table 4.3.

Table 4.4 shows the pair-wise evaluation of inter-annotator agreement of Micro-WNOp's annotators. From this analysis we can see how annotators J1 and J3 of Micro-WNOp(2) and J4 and J5 of Micro-WNOp(3) have a rather high agreement. Annotator J2 has a lower agreement with J1 and J3 and its values are comparable with those scored by SentiWordNet 1.0 on the same section (see Table 4.3).

Table 4.5 shows the agreement of the best version of SentiWordNet (1.0) and the best baseline ($\hat{\Phi}_{Objective}$) with each annotator of Micro-WNOp(2) and Micro-WNOp(3). Again SentiWordNet 1.0 obtains a relevant improvement over the $\hat{\Phi}_{Objective}$ baseline function. On Micro-WNOp(2), the values show an agreement of SentiWordNet 1.0 with annotator J1 and J3 similar to those of J2, when at the same time it obtains a lower agreement (higher MSE values) with J2. We have found a motivation of this disagreement by observing the average score assigned by the three annotators J1, J2 and J3, and SentiWordNet 1.0 to the synsets of Micro-WNOp(2), reported in Table 4.6. These averages

	Micro-WNOp	
	MSE_M	MSE_μ
$\hat{\Phi}_{Positive}$	.532	.630
$\hat{\Phi}_{Negative}$	.512	.618
$\hat{\Phi}_{Objective}$	*.462*	*.311*
SentiWordNet 1.0	**.261**	**.182**
SentiWordNet 1.1	.354	.242

Table 4.2: MSE_M and MSE_μ values of SentiWordNet 1.0 and 1.1, and the three baseline functions $\hat{\Phi}_b$, on the whole Micro-WNOp gold standard.

	Micro-WNOp(1)		Micro-WNOp(2)		Micro-WNOp(3)	
	MSE_M	MSE_μ	MSE_M	MSE_μ	MSE_M	MSE_μ
$\hat{\Phi}_{Positive}$	.536	.672	.458	.552	.590	.698
$\hat{\Phi}_{Negative}$	.545	.647	.487	.593	.538	.637
$\hat{\Phi}_{Objective}$	*.457*	*.284*	*.393*	*.280*	*.521*	*.348*
SWN 1.0	**.312**	**.202**	**.195**	**.144**	**.315**	**.216**
SWN 1.1	.391	.247	.262	.188	.436	.294

Table 4.3: MSE_M and MSE_μ values of SentiWordNet 1.0 and 1.1, and the three baseline functions $\hat{\Phi}_b$, on the three components of the Micro-WNOp gold standard.

show that J1 and J3 are centered on similar average values, while SentiWordNet 1.0 is slightly biased toward objectivity, and J2 is much more biased toward subjectivity. Thus the disagreement between SentiWordNet 1.0 and J2 is mainly generated by this “opposite” bias.

In conclusion the results indicates SentiWordNet 1.0 as a better resource than SentiWordNet 1.1, while both resources perform much better than the baseline.

4.5.1 Some statistics

Tables from 4.7 to 4.10 show some statistics about the distribution of scores in the two SentiWordNet versions. Analyzing SentiWordNet 1.0, the first remarkable fact is that the synsets judged to have some degree of opinion-related properties (i.e. not fully Objective) are a con-

A	B	MSE_M	MSE_μ
J1	J2	.157	.137
J1	J3	.056	.043
J2	J3	.184	.157
J4	J5	.036	.028

Table 4.4: Pair-wise evaluation of inter-annotator agreement of Micro-WNOp's annotators.

A	B	MSE_M	MSE_μ
J1	SentiWordNet 1.0	.204	.130
J1	$\hat{\Phi}_{Objective}$	.418	.244
J2	SentiWordNet 1.0	.382	.323
J2	$\hat{\Phi}_{Objective}$	.594	.502
J3	SentiWordNet 1.0	.156	.102
J3	$\hat{\Phi}_{Objective}$	.345	.207
J4	SentiWordNet 1.0	.331	.231
J4	$\hat{\Phi}_{Objective}$	.539	.345
J5	SentiWordNet 1.0	.328	.216
J5	$\hat{\Phi}_{Objective}$	.537	.367

Table 4.5: Agreement of SentiWordNet 1.0 and $\hat{\Phi}_{Objective}$ with each annotator of Micro-WNOp(2) and Micro-WNOp(3).

	avg(Positive)	avg(Negative)	avg(Objective)
J1	.189	.148	.664
J2	.316	.237	.447
J3	.191	.127	.683
SentiWordNet 1.0	.123	.095	.783

Table 4.6: Average scores on Micro-WNOp(2).

siderable part of the whole WordNet i.e. 24.63% of it. However, as the objectivity score decreases, indicating a stronger subjectivity score (either as Positive, or as Negative, or as a combination of them), the number of the synsets involved decreases rapidly, from 10.45% for $Obj(s) \leqq 0.5$, to 0.56% for $Obj(s) \leqq 0.125$. This seems to indicate that there are only few terms that are unquestionably Positive (or Negative), where "unquestionably" here indicates widespread agreement among different automated classifiers; in essence, this is the same observation which has independently been made in [2], where agreement among human classifiers is shown to correlate strongly with agreement among automated classifiers, and where such agreement is strong only for a small subset of "core", strongly-marked terms.

Table 4.7 reports a breakdown by POS of the scores obtained by synsets. It is quite evident that "adverb" and "adjective" synsets are evaluated as (at least partially) Subjective (i.e. $Obj(s) < 1$) much more frequently (39.66% and 35.7% of the cases, respectively) than "verb" (11.04%) or "noun" synsets (9.98%). This fact seems to indicate that, in natural language, opinions are most often conveyed by parts of speech used as modifiers (i.e. adverbs, adjectives) rather than parts of speech used as heads (i.e. verbs, nouns), as exemplified by expressions such as `a disastrous appearance` or `a fabulous game`. This intuition might be rephrased by saying that the most frequent role of heads is to denote entities or events, while that of modifiers is (among other things) to express a judgment of merit on them.

Another surprising result is that Subjective synsets related to adverbs are largely unbalanced toward the Positive dimension, while for the other POS the Subjective synsets are balanced between Positive and Negative.

Analysis of SentiWordNet 1.1, show that its values are very unbalanced toward to Objective dimension, with just a small part of synsets (9.20%) with a component of subjectivity. Most of the considerations made of SentiWordNet 1.0 still apply on this version, however with much smaller proportions. The bias toward objectivity is probably the main factor of the lower performance of SentiWordNet 1.1 on the gold standard.

Score	Positive	Negative	Objective	Positive	Negative	Objective
	Adjectives			**Verbs**		
$0 \leqq x < 0.125$	65.77%	62.81%	0.08%	89.98%	87.93%	0.00%
$0.125 \leqq x < 0.25$	12.12%	7.32%	2.14%	4.43%	4.94%	0.21%
$0.25 \leqq x < 0.375$	8.81%	8.68%	7.42%	2.66%	2.95%	0.64%
$0.375 \leqq x < 0.5$	4.85%	5.19%	11.73%	1.55%	1.81%	1.35%
$0.5 \leqq x < 0.625$	3.74%	5.63%	9.50%	0.84%	1.24%	2.67%
$0.625 \leqq x < 0.75$	2.94%	5.53%	7.65%	0.84%	1.24%	2.67%
$0.75 \leqq x < 0.875$	1.28%	3.72%	9.21%	0.10%	0.42%	4.57%
$0.875 \leqq x < 1$	0.47%	1.07%	7.57%	0.07%	0.08%	6.11%
$x = 1$	0.03%	0.04%	44.71%	0.00%	0.00%	81.05%
Avg	**0.106**	**0.151**	**0.743**	**0.026**	**0.034**	**0.940**
	Names			**Adverbs**		
$0 \leqq x < 0.125$	90.80%	89.25%	0.00%	43.70%	76.99%	0.00%
$0.125 \leqq x < 0.25$	4.53%	3.93%	0.23%	6.25%	9.66%	0.57%
$0.25 \leqq x < 0.375$	2.37%	2.42%	0.87%	6.17%	5.32%	3.00%
$0.375 \leqq x < 0.5$	1.25%	1.54%	1.84%	14.44%	2.51%	12.83%
$0.5 \leqq x < 0.625$	0.62%	1.35%	2.32%	22.63%	2.70%	23.91%
$0.625 \leqq x < 0.75$	0.24%	0.91%	2.57%	5.70%	1.72%	13.56%
$0.75 \leqq x < 0.875$	0.14%	0.48%	3.27%	1.06%	0.82%	6.11%
$0.875 \leqq x < 1$	0.05%	0.12%	5.40%	0.05%	0.27%	7.04%
$x = 1$	0.00%	0.00%	83.50%	0.00%	0.00%	32.97%
Avg	**0.022**	**0.034**	**0.944**	**0.235**	**0.067**	**0.698**

Table 4.7: Percentages of WordNet synsets that have obtained a given score in SentiWordNet 1.0 for our three categories of interest, grouped by POS, and average scores obtained for all WordNet synsets with a given POS.

Score	Positive	Negative	Objective	Positive	Negative	Objective
	Adjectives			**Verbs**		
$0 \leqq x < 0.125$	89.94%	90.14%	0.52%	99.98%	99.88%	0.05%
$0.125 \leqq x < 0.25$	3.03%	2.29%	0.95%	0.01%	0.04%	0.01%
$0.25 \leqq x < 0.375$	2.37%	1.86%	2.29%	0.00%	0.04%	0.00%
$0.375 \leqq x < 0.5$	1.74%	1.94%	2.97%	0.00%	0.01%	0.00%
$0.5 \leqq x < 0.625$	1.32%	1.61%	3.71%	0.00%	0.00%	0.01%
$0.625 \leqq x < 0.75$	0.94%	1.34%	4.27%	0.00%	0.00%	0.05%
$0.75 \leqq x < 0.875$	0.45%	0.50%	5.41%	0.00%	0.01%	0.05%
$0.875 \leqq x < 1$	0.17%	0.00%	38.10%	0.07%	0.08%	6.11%
$x = 1$	0.02%	0.04%	41.77%	0.01%	0.00%	97.48%
Avg	**0.045**	**0.050**	**0.905**	**0.001**	**0.001**	**0.998**
	Names			**Adverbs**		
$0 \leqq x < 0.125$	99.83%	99.81%	0.01%	99.51%	99.48%	0.13%
$0.125 \leqq x < 0.25$	0.07%	0.08%	0.02%	0.21%	0.24%	0.02%
$0.25 \leqq x < 0.375$	0.02%	0.02%	0.02%	0.08%	0.05%	0.02%
$0.375 \leqq x < 0.5$	0.02%	0.02%	0.02%	0.05%	0.03%	0.13%
$0.5 \leqq x < 0.625$	0.01%	0.01%	0.04%	0.05%	0.08%	0.08%
$0.625 \leqq x < 0.75$	0.01%	0.01%	0.05%	0.00%	0.03%	0.14%
$0.75 \leqq x < 0.875$	0.01%	0.01%	0.16%	0.03%	0.00%	0.46%
$0.875 \leqq x < 1$	0.01%	0.01%	3.10%	0.03%	0.03%	17.49%
$x = 1$	0.01%	0.02%	96.55%	0.03%	0.05%	81.50%
Avg	**0.001**	**0.002**	**0.997**	**0.004**	**0.006**	**0.990**

Table 4.8: Percentages of WordNet synsets that have obtained a given score in SentiWordNet 1.1 for our three categories of interest, grouped by POS, and average scores obtained for all WordNet synsets with a given POS.

Score	Positive	Negative	Objective
	All parts of speech		
$0 \leqq x < 0.125$	85.18%	84.45%	0.02%
$0.125 \leqq x < 0.25$	5.79%	4.77%	0.54%
$0.25 \leqq x < 0.375$	3.56%	3.58%	1.97%
$0.375 \leqq x < 0.5$	2.28%	2.19%	3.72%
$0.5 \leqq x < 0.625$	1.85%	2.07%	4.20%
$0.625 \leqq x < 0.75$	0.87%	1.64%	3.83%
$0.75 \leqq x < 0.875$	0.35%	1.00%	4.47%
$0.875 \leqq x < 1$	0.12%	0.27%	5.88%
$x = 1$	0.01%	0.01%	75.37%
Avg	**0.043**	**0.054**	**0.903**

Table 4.9: Scores obtained in SentiWordNet 1.0 by WordNet synsets (all parts of speech considered altogether).

Score	Positive	Negative	Objective
	All parts of speech		
$0 \leqq x < 0.125$	98.24%	98.25%	0.10%
$0.125 \leqq x < 0.25$	0.55%	0.43%	0.16%
$0.25 \leqq x < 0.375$	0.40%	0.32%	0.38%
$0.375 \leqq x < 0.5$	0.29%	0.33%	0.49%
$0.5 \leqq x < 0.625$	0.22%	0.27%	0.63%
$0.625 \leqq x < 0.75$	0.15%	0.22%	0.73%
$0.75 \leqq x < 0.875$	0.08%	0.08%	1.00%
$0.875 \leqq x < 1$	0.03%	006%	9.09%
$x = 1$	0.01%	0.00%	87.37%
Avg	**0.008**	**0.009**	**0.983**

Table 4.10: Scores obtained in SentiWordNet 1.1 by WordNet synsets (all parts of speech considered altogether).

4.6 Conclusions

We have presented SentiWordNet an automatically generated lexical resource in which each WordNet synset is tagged with a triplet of numerical scores representing how Positive, Negative, and Objective a synset is. We have defined and evaluated two versions of SentiWordNet, 1.0 and 1.1. Both versions have produced a relevant improvement over the baseline, with SentiWordNet 1.0 showing the best results. We have presented results an from inter-annotators agreement study, and compared them with SentiWordNet, showing that, while there is still room for improvement, SentiWordNet data, especially in version 1.0, can be considered of good quality.

SentiWordNet can prove a useful tool for opinion mining applications, because of its wide coverage (*all* WordNet synsets are tagged according to *each* of the three labels Objective, Positive, Negative) and because of its fine grain, obtained by qualifying the labels by means of numerical scores.

To be used in its full capabilities, a resource like SentiWordNet obviously requires that the text on which it is used have to be disambiguated, with each occurrence of any term assigned to the WordNet synset it belongs. This requires to perform an accurate WSD process on text. At the current time WSD technology can guarantee at best a 80% level of accuracy [48], which can produce a "chain effect" on the quality of results obtained by using SentiWordNet, making it less effective than a term-based lexical resource. However, this doesn't make SentiWordNet less useful, for at least two reasons:

- SentiWordNet has been already successfully used in opinion related tasks, without the use of WSD tools. For example, Attardi and Simi [5], have extracted a set of 8,427 "opinionated word" from SentiWordNet just by collapsing all the score triplets for all the senses of a term in a single triplet, and selecting as opinionated those terms whose sum of positive and negative score is above 0.4. Such resource has been used to tag opinionated words into the TREC Blog06 [74] corpus, and has produced, on the retrieval system the authors used to participate to the 2006 TREC Blog track, a 11.6% improvement in precision with respect to the system not using it.

- we have separated two problems: disambiguating term senses, and determining opinion-related properties of a term sense. We think that this would help research to focus on better defined problems. As also Ide states in [48], we can expect that WSD technology will improve in the next years, reaching an higher level of accuracy. At that time SentiWordNet[12] would be a good companion for a WSD tool to perform opinion related analysis of text.

[12] Which hopefully will improve too...

Chapter 5

Random-Walk Models of Term Semantics

Abstract

This chapter presents an application of three random-walk models to ranking WordNet synsets in terms of how strongly they possess a given semantic property. The random-walk models are based on PageRank, the well-know random-walk algorithm originally devised for ranking Web search results. The rationale of applying a PageRank-like algorithm to detecting the semantic properties of synsets lies in the fact that the space of WordNet synsets may be seen as a graph, in which synsets are connected, for example, through the binary relation (denoted by $s_i \blacktriangleright s_k$) "a term belonging to synset s_k occurs in the gloss of synset s_i", and through which the observed properties "flow".

We explore also random-walk models for two other properties which may be equally adequate to this task: a first variation which is based on the "inverse" relation $s_i \blacktriangleleft s_k$, i.e., with properties flowing from the *definiens* to the *definiendum*, and a second one based on the bidirectional relation $\blacktriangleleft\!\blacktriangleright$, which assumes that properties may flow from the *definiens* to the *definiendum* and viceversa. We report experimental results supporting our intuitions. We also produce a new version of SentiWordNet (2.0) which improves the version 1.0 presented in Chapter 4.

5.1 Introduction

In this chapter we present a novel method for ranking the *entire* set of WordNet synset, according to a given semantic property. Following the main subject of this thesis we focus on opinion-related properties (ORPs). Specifically, we experiment on ranking synsets by orientation (though we deem that the proposed method can be applied to many other dimensions of term semantics). Two rankings are produced, one according to positivity and one according to negativity. Note that it is *not* the case that one is the inverse of the other, since e.g., the least positive synsets may be negative or objective synsets alike; in fact, we obtain the two rankings independently of each other.

The main idea underlying our ranking method is that the positivity and negativity of WordNet synsets can be determined by mining their glosses. The idea of using the information contained in the glosses to determine ORPs is one of the "building blocks" of this thesis; however, we would like to point out that in this chapter we make a completely different use of such resource with respect to previous chapters. In this case our idea crucially relies on the observation that the gloss of a WordNet synset contains terms, and on the hypothesis that the glosses of positive (resp. negative) synsets will mostly contain terms belonging to positive (negative) synsets. This means that the binary relation $s_i \blacktriangleright s_k$ ("the gloss of synset s_i contains a term belonging to synset s_k"), which induces a directed graph on the set of WordNet synsets, may be thought of as a channel through which positivity and negativity flow, from the *definiendum* (the synset s_i being defined) to the *definiens* (the synsets s_k that contribute to the definition of s_i by virtue of their member terms occurring in the gloss of s_i). In other words, if a synset s_i is known to be positive (negative), this can be viewed as an indication that the synsets s_k to which the terms occurring in the gloss of s_i belong, are themselves positive (negative). We define this ORP flow model as the *direct flow* model.

The fact that the $\blacktriangleright$ relation is not explicit in WordNet is circumvented by actually using eXtendedWordNet [43], a publicly available, automatically sense-disambiguated version of WordNet in which every term occurring in a gloss is replaced by the synset it is deemed to belong to.

The two other random-walk models, that we illustrate in the chapter, may also be plausible choices for controlling the logic of ORP flow.

The second is a random-walk model different from the direct-flow model, which is applied to the "inverse" graph, i.e., the graph defined by the binary relation $s_i \blacktriangleleft s_k$ ("a term belonging to synset s_i appears in the gloss of synset s_k") with properties flowing from the *definiens* to the *definiendum* (*inverse flow*). The third model is a bidirectional random-walk model based on the binary relation $\blacktriangleleft\!\!\blacktriangleright$, which assumes that properties may flow from the *definiens* to the *definiendum* and viceversa (*bidirectional flow*).

We show how the three mentioned flow models can be implemented by the three random-walk algorithms based on the well-known PageRank algorithm [13]. We like to point out that the choice of the PageRank algorithm has not been made *a-priori* but instead comes after the definition of the three ORP flow models, as the fact that the three PageRank-based algorithms result to be the correct realizations of the models.

PageRank, a random-walk algorithm for ranking Web search results which lies at the basis of the Google search engine, is probably the most important single contribution to the fields of information retrieval and Web search of the last ten years, and was originally devised in order to detect how authoritativeness flows in the Web graph and how it is conferred onto Web sites. The advantages of PageRank are its strong theoretical foundations, its fast convergence properties, and the effectiveness of its results. The reason why PageRank (and the PageRank-like variants we propose), among all random-walk algorithms, is particularly suited to our application will be discussed in the rest of the chapter.

We report results which compare the three algorithms on the task of producing positivity and negativity rankings of WordNet synsets. Note however that our method is not limited to ranking synsets *by positivity or negativity*, and can in principle be applied to the determination of other semantic properties of synsets, such as membership in a domain, since for many other properties we may hypothesize the existence of a similar "hydraulics" between synsets. We thus see positivity and negativity only as proofs-of-concept for the potential of the method.

The result of applying a PageRank-like algorithm to WordNet synsets is just a ranked list of all the WordNet synsets, where a synset s_i can be consider more positive[1] (or negative) than any synset with a lower rank position, and less positive (or negative) than any synset with

[1] The rank positions are determined by numeric values assigned to each synset by the algorithm, with the possibility of *ties*, i.e., synsets with the same score assigned.

a higher rank position. The actual numeric scores assigned by PageRank (and the two variants) are not directly usable to assign to synsets a SentiWordNet-like score. In the final part of this chapter we present a simple yet effective way to produce from the best performing rankings of our experiments a new version of SentiWordNet (2.0), which improves on the version 1.0 presented in Chapter 4.

5.1.1 Chapter outline

Section 5.2 reports on related work, focusing on the use of random-walk models to perform term sense-related tasks. In Section 5.3 we describe the PageRank-based model of ORP flow. In Section 5.4 we present our modifications of this model, resulting in two random-walk models each departing from the purely PageRank-based in a different direction. Section 5.5 describes the structure of our experiments and Section 5.6 discusses the results we have obtained. In Section 5.7 we describe how we have derived a new version of SentiWordNet (2.0) from the best results of the experiments, and we compare it with version 1.0. Section 5.8 concludes.

5.2 Related work

In this related work section we will focus on works that have used random-walk models to perform term sense-related tasks. We point the reader to Chapter 4, Section 4.2, for a general discussion on related works on ORPs of term senses.

5.2.1 Mihalcea et al.

Mihalcea et al. [67] have proposed a word sense disambiguation (WSD) method based on the use of PageRank.

Given a sentence in which some of the terms are ambiguous, i.e., have multiple senses, the WSD method proposed by Mihalcea et al. consists in the following steps:

1. The text to be disambiguated is tokenized and POS-tagged.

2. A graph $G = \langle N, L \rangle$ is built, first adding to the node set N all the possible senses for all the terms in the sentence (including non-

ambiguous ones), using information from POS-tagging to limit the number of nodes added.

3. G is expanded adding links in L and nodes in N by navigating many of the semantic relations of WordNet (e.g., hypernymy, hyponymy).

4. PageRank is executed on G (see Section 5.3.1 for details), obtaining a ranked list of all the nodes in N.

5. For each ambiguous term, the highest ranked of its senses is assigned as the intended sense.

The WSD task is obviously different from the one we face in this chapter, but the real key difference between the method of Mihalcea et al. [67] and ours is that their method is designed to work on an *case-by-case problem*, i.e. the disambiguation of three sentences require three distinct executions of the program, each one based on the creation of a local graph to be given in input to PageRank. We instead work on a *global problem*, i.e., the evaluation of some semantic properties on *all*[2] the elements of the language.

5.2.2 Hughes and Ramage

Hughes and Ramage [47] have recently proposed, independently from our work, the use of random-walk models to determine the semantic relatedness between WordNet synsets.

The graph $G = \langle N, L \rangle$ they build contains, as nodes, all the WordNet synsets (e.g. [`dog(N,3)`]), terms with POS (e.g. `dog(n)`) and just terms (e.g. `dog`). The links of the graph are defined by the various WordNet relations (e.g. hypernymy, hyponymy), links between synsets, terms with POS and terms, and also a gloss-based relation. The gloss-based relation is similar to our direct flow ▶ relation, although it does not uses a sense-disambiguated version of WordNet glosses, but creates instead links between the synset and the terms with POS appearing in its gloss, and then from each term with POS to all the synsets containing it.

They propose to measure the semantic relatedness between two synsets s_i and s_j by producing two customized ranking r_i and r_j of all the nodes

[2] At least, all those represented in WordNet.

N by using PageRank on the graph G and giving in input a *personalization vector* (see Section 5.3.1) where only the value related to s_i (or s_j) is non-null. Then they use a Zero-KL divergence measure to measure the similarity between the two rankings.

Their work have thus some points in common with ours, although they do not provide a clear motivation on why the PageRank algorithm is the right implementation of their random-walk model.

5.3 The PageRank model of ORP flow

5.3.1 PageRank

Let $G = \langle N, L \rangle$ be a directed graph, with N its set of nodes (e.g. Web documents) and L its set of directed links (e.g. Web links); let $\mathbf{W}_0$ be the $|N| \times |N|$ *adjacency matrix* of G, i.e., the matrix such that $\mathbf{W}_0[i,j] = 1$ iff there is a link from node n_i to node n_j. We will denote by $B(i) = \{n_j \mid \mathbf{W}_0[j,i] = 1\}$ the set of the *backward neighbors* of n_i, which are connected to n_i by the *backward links* of n_i, and by $F(i) = \{n_j \mid \mathbf{W}_0[i,j] = 1\}$ the set of the *forward neighbors* of n_i, which are connected to n_i by the *forward links* of n_i. Let $\mathbf{W}$ be the *row-normalized adjacency matrix* of G, i.e., the matrix such that $\mathbf{W}[i,j] = \frac{1}{|F(i)|}$ iff $\mathbf{W}_0[i,j] = 1$ and $\mathbf{W}[i,j] = 0$ otherwise.

The input to PageRank is the row-normalized adjacency $\mathbf{W}$ matrix (plus a *personalization vector* $\mathbf{e}$ to be discussed later), and its output is a vector $\mathbf{a} = \langle a_1, \ldots, a_{|N|} \rangle$, where a_i represents the "score" of node n_i, which in our application measures the degree to which n_i has the ORP of interest. PageRank iteratively computes $\mathbf{a}$ based on the formula

$$a_i^{(k)} \leftarrow \alpha \sum_{j \in B(i)} \frac{a_j^{(k-1)}}{|F(j)|} + (1-\alpha) e_i \tag{5.1}$$

where $a_i^{(k)}$ denotes the value of a_i, the i-th entry of vector $\mathbf{a}$, at the k-th iteration, e_i is a constant such that $\sum_i e_i = 1$, and $0 \leq \alpha \leq 1$ is a control parameter. In vectorial form, Equation 5.1 can be written as

$$\mathbf{a}^{(k)} = \alpha \mathbf{a}^{(k-1)} \mathbf{W} + (1-\alpha)\mathbf{e} \tag{5.2}$$

The underlying intuition is that a node n_i has a high score when (recursively) it has many high-scoring backward neighbors; a node n_j

thus passes its score a_j along to its forward neighbors $F(j)$, but this score is subdivided equally among the members of $F(j)$, so that the higher $|F(j)|$ is, the smaller is the score that n_j contributes to each of the nodes in $F(j)$. This mechanism (that is represented by the summation in Equation 5.1) is then "smoothed" by the e_i constants.

This is called a *random-walk* model [69] since a_i can be seen as the expected frequency with which a "random walker", after starting from a node n_k picked at random with probability e_k, visits node n_i during an infinite walk through G in which, once at node n_j, he/she follows one of the $|F(j)|$ forward links of n_j each with probability $\frac{\alpha}{|F(j)|}$, or jumps to some random node n_l with probability $(1-\alpha)e_l$.

Vector $\mathbf{a}$, the fixpoint of Equation 5.2, corresponds to the stationary distribution of the Markov chain associated to the random walk, i.e., to the *principal eigenvector* of the $\alpha\mathbf{W}+(1-\alpha)\mathbf{E}\times\mathbf{1}$ matrix (where $\mathbf{1}$ is a vector of all ones) defined as the eigenvector associated to the eigenvalue with the largest magnitude[3].

In the original application of PageRank for ranking Web search results the elements of $\mathbf{e}$ are usually taken to be all equal to $\frac{1}{|N|}$, thus modeling a random walker who may jump to any Web page with equal probabilities. However, it is possible to give different values to different pages in $\mathbf{e}$. In fact, the value of e_i amounts to an *internal source of score* for n_i that is constant across the iterations and independent from its backward neighbors. For instance, attributing a null e_i value to all but a few Web pages characterized by a given topic can be used in order to bias the ranking of Web pages in favor of this topic [46].

5.3.2 Ranking WordNet synsets with PageRank

Our first model of ORP flow is based on a rather straightforward mapping of the Web model to the WordNet model: Web documents are substituted by WordNet synsets and Web links are substituted by the ▶ relation. Two different and independent rankings are produced, one for positivity and one for negativity. The e_i values are used as internal sources of positivity (resp. negativity) by attributing a null e_i value to all but a few "seed" synsets of renowned positivity (negativity). Through its iterations PageRank will thus make positivity (negativity) flow from

[3] The computational properties of the PageRank algorithm, and how to compute it efficiently, have been widely studied; the interested reader may consult [10].

the seed synsets, from which positivity flows out at a rate constant throughout the iterations, into other synsets along the $\blacktriangleright$ relation (by using the $\mathbf{W}^{\blacktriangleright}$ matrix defined on this relation), until a stable state is reached; at this point the a_i values can be used to rank the synsets in terms of positivity (negativity).

In details, our algorithm consists in the following steps:

1. The graph $G^{\blacktriangleright} = \langle N, L^{\blacktriangleright} \rangle$ on which PageRank will be applied is generated. We define N to be the set of all WordNet synsets; in WordNet 2.0 there are 115,424 of them. We define L to contain a link from synset s_i to synset s_k iff the gloss of s_i contains *at least* a term belonging to s_k (terms occurring in the examples phrases are not considered), based on the disambiguation information contained in eXtendedWordNet(see Section 5.5.1). Numbers, articles and prepositions occurring in the glosses are discarded, since they can be assumed to carry no positivity and negativity, and since they do not belong to a synset of their own. This leaves only nouns, adjectives, verbs, and adverbs.

2. The graph $G^{\blacktriangleright} = \langle N, L^{\blacktriangleright} \rangle$ is "pruned" by removing "self-loops", i.e., links going from a synset s_i into itself (since we assume that there is no flow of semantics from a concept unto itself). The row-normalized adjacency matrix $\mathbf{W}^{\blacktriangleright}$ of $G^{\blacktriangleright}$ is derived.

3. We load the e_i values into the $\mathbf{e}$ vector; all synsets other than the seed synsets of renowned positivity (negativity) are given a value of 0. We experiment with several different versions of the $\mathbf{e}$ vector; see Section 5.5.4 for details.

4. PageRank is executed using $\mathbf{W}^{\blacktriangleright}$ and $\mathbf{e}$, iterating until a predefined termination condition is reached. The termination condition we use in this paper consists in checking that the cosine of the angle between the vectors $\mathbf{a}^{(k)}$ and $\mathbf{a}^{(k+1)}$ generated by two subsequent iterations is over a predefined threshold $1 - \chi$ (we use a value of $\chi = 10^{-9}$).

5. We rank all the synsets of WordNet in descending order of their a_i score.

The process is run twice, once for positivity and once for negativity. The only difference between the two runs is in Step 3, since two different vectors, one of positive and the other of negative seed synsets, are used in the two runs.

5.3.3 Why PageRank?

The use of the PageRank to implement the direct flow model seems a reasonable choice, because many interesting intuition are represented into the PageRank formula:

1. If terms contained in synset s_k occur in the glosses of many positive synsets, and if the positivity scores of these synsets are high, then it is likely that s_k is itself positive (the same happens for negativity). This justifies the summation of Equation 5.1.

2. If the gloss of a positive synset that contains a term in synset s_k also contains many other terms, then this is a weaker indication that s_k is itself positive (this justifies dividing by $|F(j)|$ in Equation 5.1).

5.4 Alternatives to the PageRank model

The *direct flow* model and its intuitions are based on the $\blacktriangleright$ relation. However, nothing prevents us from considering alternative relations.

5.4.1 The inverse flow model

Starting from the same intuition of the direct flow model it is equally plausible to hypothesize an *inverse flow* model, in which the synsets that occur in the gloss of the *definiendum* influence the *definiendum* itself, and not viceversa. In this model the ORP thus flows from the *definiens* to the *definiendum*, along the $\blacktriangleleft$ relation (defined as the symmetric relation of $\blacktriangleright$).

We formalize the inverse flow model by the equation

$$a_i^{(k)} \leftarrow \frac{\alpha}{|B(i)|} \sum_{j \in B(i)} a_j^{(k-1)} + (1-\alpha)e_i \tag{5.3}$$

where $B(i)$ is now derived from the adjacency matrix $\mathbf{W}_\mathbf{0}^{\blacktriangleleft}$ defined by the $\blacktriangleleft$ relation.

We stress that the inverse flow model is characterized not only by a different incidence matrix with respect to the direct flow model, but by a very different equation of the "hydraulics" of ORP flow. In fact, Equation 5.3 states that node a_i receives the *average*, and not the sum, of the scores of the nodes that point to a_i, modulo α and e_i. In the case of inverse flow we consider this a reasonable assumption since:

1. If the gloss of a synset s_k contains many terms that belong to positive synsets, and if the positivity scores of these synsets are high, then it seems likely that s_k is itself positive (the same happening for negativity), which justifies the summation of Equation 5.3.

2. If the gloss of a synset s_i that contains a term belonging to a positive synset s_k also contains many other terms, this seems a weaker indication that s_i is itself positive (which justifies dividing by $|B(i)|$ in Equation 5.3).

In order to write Equation 5.3 in matrix form we may exploit the fact that $\mathbf{W}_\mathbf{0}^{\blacktriangleleft}$ happens to be equal to $(\mathbf{W}_\mathbf{0}^{\blacktriangleright})^T$, the transpose of $\mathbf{W}_\mathbf{0}^{\blacktriangleright}$, and that applying the normalization factor $|B(i)|$ in Equation 5.3 is equivalent to performing column normalization on $\mathbf{W}_\mathbf{0}^{\blacktriangleleft}$. Thus $\mathbf{W}^{\blacktriangleleft} = (\mathbf{W}^{\blacktriangleright})^T$, and Equation 5.3 can be written in matrix form as

$$\mathbf{a}^{(k)} = \alpha\mathbf{a}^{(k-1)}(\mathbf{W}^{\blacktriangleright})^T + (1-\alpha)\mathbf{e} \tag{5.4}$$

where $\mathbf{W}^{\blacktriangleright}$ is the row-normalized adjacency matrix used for the direct flow model in Equation 5.2. This indicates that, even if Equation 5.3 is very different from the equation that originates PageRank (Equation 5.1), the inverse flow model can anyway be computed by using PageRank, with the only difference that the $\mathbf{W}^{\blacktriangleright}$ matrix of the direct model needs to be replaced by its transpose $(\mathbf{W}^{\blacktriangleright})^T$.

5.4.2 The bidirectional model

We have argued that both the direct flow and the inverse flow models are reasonable models of how ORPs flow between synsets. Actually, in this analysis no argument has been put forward that either model is better than the other, or that the two models are mutually incompatible. It seems thus plausible that a third, *bidirectional flow* model could be hypothesized, in which ORPs flow from the *definiendum* to the *definiens* and vice versa, pretty much as in an electrical network. A synset s_k is thus seen to distribute its positivity score both to the synsets which occur in its gloss (the $\blacktriangleright$ relation) and to the synsets whose glosses contain it (the $\blacktriangleleft$ relation). The binary relation $\blacktriangleleft\!\!\blacktriangleright$ according to which ORPs flow in the bidirectional model is thus defined as $\blacktriangleleft\!\!\blacktriangleright \equiv \blacktriangleleft \cup \blacktriangleright$. we formalize the bidirectional flow model by the following equation:

$$\begin{aligned} a_i^{(k)} \quad \leftarrow \quad & \alpha \sum_{j \in B^{\blacktriangleright}(i)} \frac{a_j^{(k-1)}}{|F^{\blacktriangleright}(j)|} + \\ & + \frac{\alpha}{|B^{\blacktriangleleft}(i)|} \sum_{j \in B^{\blacktriangleleft}(i)} a_j^{(k-1)} + (1-\alpha) e_i \end{aligned} \tag{5.5}$$

where the $B^{\blacktriangleright}$, $F^{\blacktriangleright}$, $B^{\blacktriangleleft}$ and $F^{\blacktriangleleft}$ are the neighborhood functions of the direct and inverse flow models.

The vectorial form of Equation 5.5 can be easily derived by observing that the normalized matrix $\mathbf{W}^{\blacktriangleleft\!\!\blacktriangleright}$ for the bidirectional flow model induced by Equation 5.5 is equal to $\mathbf{W}^{\blacktriangleright} + (\mathbf{W}^{\blacktriangleright})^T$; we thus obtain

$$\mathbf{a}^{(k)} = \alpha \mathbf{a}^{(k-1)} (\mathbf{W}^{\blacktriangleright} + (\mathbf{W}^{\blacktriangleright})^T) + (1-\alpha)\mathbf{e} \tag{5.6}$$

Again, this formula shows that also the bidirectional flow model can be computed using PageRank, with the only difference that the $\mathbf{W}^{\blacktriangleright} + (\mathbf{W}^{\blacktriangleright})^T$ matrix needs to be used in place of the $\mathbf{W}^{\blacktriangleright}$ matrix of the direct model and of the $(\mathbf{W}^{\blacktriangleright})^T$ of the inverse model.

The reasons why the bidirectional flow model can be a good model of ORP flow follow from those discussed for the direct flow and inverse flow models in Sections 5.3 and 5.4.1, given that this model is just the union of the two models, with no modification to their specific characteristics.

5.5 Experiments

5.5.1 WordNet and eXtendedWordNet

WordNet glosses are made of simple text, for human reading, and are not intended for automatic processing by algorithms[4]. The most relevant

[4] Fortunately, this consideration may be no more valid in the future, since there is a on-going project to produce a synset-level manually disambiguated version of WordNet glosses (version 3.0).

aspect for us is that a gloss does not contain any information regarding the intended sense (expressed with a synset) of the terms appearing in it. Thus, the transformation of WordNet into a graph based on the ▶ relation (or any of the other proposed relations) would of course be non-trivial:

- Synsets only contain lemmas, while different inflected forms of a lemma may occur in the glosses. A lemmatization step would thus be necessary in order to map all the terms appearing in the glosses into their corresponding lemmas.
- There may be several synsets containing the same pair $\langle l_k, POS_k \rangle$ (e.g. $\langle$`bank`, $N\rangle$ has several senses, hence it appears in several synsets), which means that a single occurrence of pair $\langle l_k, POS_k \rangle$ in the gloss of a synset s_i might generate several links outgoing from s_i, each one incoming into each synset containing $\langle l_k, POS_k \rangle$. This is problematic, since it means that spurious score transfers would take place, to unintended senses of a term s_k.

In order to avoid the two problems mentioned above we use eXtendedWordNet [43], a publicly available version of WordNet in which (among other things) each term s_k occurring in a WordNet gloss (except those in example phrases) is lemmatized and semantically disambiguated, i.e., mapped to the synset in which it belongs[5]. We have used eXtendedWordNet version 2.0-1.1, which refers to WordNet version 2.0. The eXtendedWordNet resource has been automatically generated, which means that the associations between terms and synsets are likely to be sometimes incorrect, and this of course introduces noise in the our method. However, this noise is obviously inherent in the fully automatic nature of our method, and on the fact that, at the time of writing, no manually disambiguated version of WordNet exists. The fact that we use eXtendedWordNet will allow us to say "synset s_k occurs in the gloss of synset s_i" when we actually mean "term s_k, which eXtendedWordNet maps to synset s_k, occurs in the gloss of synset s_i".

Figure 5.1 displays the five WordNet synsets that contain the lemma `tidy` (Column 1), together with their WordNet glosses (Column 2), and eXtendedWordNet glosses (Column 3).

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

Synset	WordNet gloss	eXtendedWordNet gloss
[tidy(N,1)]	receptacle that holds odds and ends (as sewing materials)	[receptacle(N,1)] that [hold(V,8)] [odds_and_ends(N,1)]
[tidy(V,1)]	put (things or places) in order; "Tidy up your room!"	[put(V,1)] ([thing(N,1)] or [place(N,6)]) in [order(N,15)]
[tidy(A,1)]	marked by good order and cleanliness in appearance or habits; "a tidy person"; "a tidy house";	[mark(V,3)] by [good(A,1)] [order(N,2)] and [cleanliness(N,2)] in [appearance(N,3)] or [habit(N,2)]
[tidy(A,2)]	(of hair) neat and tidy; "a nicely kempt beard"	(of [hair(N,1)]) [neat(A,1)] and [tidy(A,2)]
[tidy(A,3)]	large in amount or extent or degree; "it cost a considerable amount"; "a goodly amount"; "received a hefty bonus"; "a respectable sum"; "a tidy sum of money"; "a sizable fortune"	[large(A,1)] in [amount(N,1)] or [extent(N,2)] or [degree(N,2)]

Figure 5.1: All the WordNet synsets containing the lemma `tidy`, together with their WordNet glosses and their sense-disambiguated eXtendedWordNet glosses.

5.5.2 The gold standard

In order to evaluate the quality of the rankings produced by our three alternative random-walk models we have used the Micro-WNOp corpus [15] as a gold standard[6], already described in Chapter 4, Section 4.4.1.

We have obtained two reference positivity (and two for negativity) rankings for our experiments from Micro-WNOp by averaging the positivity (negativity) scores assigned to every synset by the evaluators into a single score, and by sorting the synsets of the two largest parts according to the resulting score. We have used the ranking derived from the Micro-WNOp(2) part of the corpus to optimize the α parameter of the PageRank algorithm. Then we have run our final experiments, using the optimized *alpha* value, on the ranking derived from the Micro-WNOp(3) part of the corpus, on which we have computed the effectiveness measure.

5.5.3 The effectiveness measure

A ranking $\prec$ is a partial order on a set N objects (synsets, in our case) $\{o_1 \ldots o_{|N|}\}$. Given a pair (o_i, o_j) of objects, o_i may precede o_j $(o_i \prec o_j)$, it may follow o_i $(o_i \succ o_j)$, or it may be tied with o_j $(o_i \approx o_j)$.

To evaluate the rankings produced by in the experiments we have used the *p-normalized Kendall τ distance* (noted τ_p – see e.g., [38]) between the Micro-WNOp rankings and those produced by the experiments. The τ_p distance, a standard function for the evaluation of rankings with ties, is defined as

$$\tau_p = \frac{n_d + p \cdot n_u}{Z} \tag{5.7}$$

where n_d is the number of *discordant pairs*, i.e., pairs of objects ordered one way in the gold standard and the other way in the tested ranking; n_u is the number of pairs which are ordered (i.e., not tied) in the gold standard and are tied in the tested ranking, and p is a penalization to be attributed to each such pair; and Z is a normalization factor (equal to the number of pairs that are ordered in the gold standard) whose aim is to make the range of τ_p coincide with the $[0, 1]$ interval. Note that pairs tied in the gold standard and ordered in the tested ranking are not considered in the evaluation.

[6] `http://www.unipv.it/wnop/`

The penalization factor is set to $p = \frac{1}{2}$, which is equal to the probability that a ranking algorithm correctly orders the pair by random guessing; there is thus no advantage to be gained from either random guessing or assigning ties between objects. For a test ranking which perfectly coincides with that from the gold standard, τ_p equals 0; for a test ranking which is exactly the inverse of the one from the gold standard, τ_p is equal to 1; for a test ranking consisting of all ties, τ_p equals $\frac{1}{2}$. The value of $\frac{1}{2}$ may thus be taken as the performance of the "trivial ranker", which must thus be improved upon by any non-trivial ranking algorithm.

5.5.4 The e vector

In order to produce a ranking by positivity (negativity) it is mandatory to provide an **e** vector as input to the random-walk algorithm (i.e. PageRank or one of the two variations), which points out which are the "sources of positivity (negativity), which then the algorithm will use to let flow the positivity (negativity) into the links graph. We have experimented with several different definitions of **e**, each for both positivity and negativity.

First of all, we have tested, as the **e** vector, a vector (dubbed **e**1) with all values uniformly set to $\frac{1}{|N|}$. This is the **e** vector that was originally used in [13] for the Web page ranking problem, and brings about an unbiased (that is, with respect to particular properties) ranking of WordNet. Of course, it is not meant to be used for ranking by positivity or negativity; we have used it simply in order to evaluate the impact of property-biased vectors for positivity (negativity) ranking.

The first sensible, minimalistic definition of **e** (hereafter dubbed **e**2) we have used is that of a vector with uniform non-null e_i scores assigned to the synsets that contain the adjective `good` (`bad`), and null scores for all other synsets. A further, still fairly minimalistic definition we have used (dubbed **e**3) is that of a vector with uniform non-null e_i scores assigned to the synsets that contain at least one of the seven "paradigmatic" positive (negative) adjectives used as seeds in [92].

We have also tested more complex versions of **e**, with e_i scores obtained from release 1.0 of SentiWordNet, described in Chapter 4. We produced an **e** vector (dubbed **e**4) in which the score assigned to a synset is proportional to the positivity (negativity) score assigned to it by SentiWordNet, and in which all entries sum up to 1. In a similar way we have also produced a further **e** vector (dubbed **e**5) through the scores of the release 1.1 of SentiWordNet.

Note that PageRank (and also the two variations) is parametric on α, which determines the balance between the contributions of the $\mathbf{a}^{(k-1)}$ vector and the **e** vector. A value of $\alpha = 0$ makes the $\mathbf{a}^{(k)}$ vector coincide with **e**, and corresponds to discarding the contribution of the random-walk algorithm. Conversely, setting $\alpha = 1$ corresponds to discarding the contribution of **e**, and makes $\mathbf{a}^{(k)}$ uniquely depend on the topology of the graph; the result is an "unbiased" ranking. The desirable cases are, of course, in between. As first hinted in Section 5.5.2, we thus optimize the α parameter on the synsets in Group1, and then test the algorithm with the optimal value of α on the synsets in Group2. All the 101 values of α from 0.0 to 1.0 with a step of .01 have been tested in the optimization phase.

5.6 Results

Table 5.1 shows the results obtained by our three models with the different choices for the **e** vector as detailed in Section 5.5.4. PageRank is iterated until the cosine of the angle between the vectors $\mathbf{a}^{(k)}$ and $\mathbf{a}^{(k+1)}$ generated by two subsequent iterations is above a predefined threshold $1 - \chi$ (we use $\chi = 10^{-9}$). However, in order to limit the amount of processing, we stop PageRank whenever this condition has not been reached in 1000 iterations.

The results indicate the performance obtained on the test set (Micro-WNOp(3) rankings) with the value of α that was determined optimal by experimentation on the validation set (Micro-WNOp(2) rankings); different values of α may thus be used for different choices of **e**. The "B" (baseline) column contains the values of τ_p as computed directly on the **e** vector, i.e., before the application of PageRank. The Δ values shown to the right of each column denote the relative improvement obtained by the method indicated against the baseline (since low values of τ_p are better, an improvement is indicated by a negative value).

Table 5.1 clearly indicates that the inverted flow model always produces the best results, irrespectively of the choice of the **e** vector. Moreover, the best absolute values for positivity (0.292) and negativity (0.222) show a large improvement with respect to their original **e** vectors (−16.4% for positivity and −25.0% for negativity). This is relevant, since they

Table 5.1: τ_p values obtained by the three proposed models; ▶, ◀ and ◀▶ indicate the direct, inverse, and bidirectional models, respectively; Δ indicates the improvement of each model with respect to the baseline ("B"), consisting of the ranking obtained by the corresponding **e** vector before the application of any ORP flow algorithm. Boldface indicates the best result obtained.

Ranking by positivity							
e	B	▶	Δ	◀	Δ	◀▶	Δ
e1	0.500	0.496	-0.8%	0.479	-4.2%	0.489	-2.1%
e2	0.500	0.467	-6.7%	0.435	-13.0%	0.457	-8.7%
e3	0.500	0.471	-5.8%	0.424	-15.1%	0.477	-4.7%
e4	0.349	0.325	-6.8%	**0.292**	-16.4%	0.312	-10.7%
e5	0.400	0.380	-4.9%	0.345	-13.6%	0.374	-6.4%
e6	–	0.292	0%	0.318	-2.1%	–	–

Ranking by negativity							
e	B	▶	Δ	◀	Δ	◀▶	Δ
e1	0.500	0.549	9.8%	0.461	-7.7%	0.506	1.2%
e2	0.500	0.502	0.3%	0.416	-16.8%	0.475	-5.1%
e3	0.500	0.495	-0.9%	0.387	-22.7%	0.452	-9.5%
e4	0.296	0.284	-4.3%	**0.222**	-25.0%	0.248	-16.4%
e5	0.407	0.393	-3.5%	0.270	-33.6%	0.319	-21.7%
e6	–	0.222	0%	0.241	-15.1%	–	–

have been obtained with vectors **e**4 (the ones derived from SentiWordNet 1.0); in other words, the improvement is obtained with respect to an already high-quality lexical resource for ORPs, obtained by the same techniques that, at the term level, are still the best-known performers for polarity detection on the widely used General Inquirer benchmark.

Although the direct flow model also improves over the baseline, the inverted flow model is largely superior to it (the latter improving on the former by 10.1% on positivity and by 21.8% on negativity). Concerning the bidirectional flow model, while it also outperforms the direct flow model, it does so less markedly than the inverse model does; in the light of the previously discussed results this is unsurprising, given that it is a combination of the other two models.

The superiority of the inverse flow model is also apparent from the results of the **e**6 experiments. Here, the inverse flow model as applied to the best vector resulting from the direct flow model manages to improve the quality of this vector (by 2.1% on positivity and by 15.1% on negativity), but still underperforms with respect to the best result it has obtained (on **e**4). On the contrary, the direct flow model as applied to the best vector resulting from the inverse flow model leaves the vectors unchanged. A closer inspection of this latter result shows that the value of α that performed optimally in this case was $\alpha = 0$, which corresponds to ... leaving the **e** vector unchanged, i.e., renouncing to let ORPs flow through the network. All values of $\alpha > 0$ managed instead to obtain an *inferior* performance with respect to the best performance obtained by the inverse model.

5.6.1 Anecdotal evaluation

Tables 5.2 and 5.3 display the ten top-ranked synsets returned by each model according to positivity and negativity. Although the tables contain just a small sample of the 115,424 WordNet synsets that are ranked by PageRank, they give interesting hints as to how the three models perform on WordNet.

The analysis of the top-ranked synsets returned by each model according to positivity and negativity shows that some of the top-ranked synsets for the direct flow model, especially for the ranking by positivity, contain function words, such as the verbs "to be" and "to have", or words that simply occur frequently within glosses, such as "quality" or "capable". These synsets receive many incoming links in the direct flow

Table 5.2: The ten top-ranked positive synsets produced by each of the three proposed models. Only the first three lemmas for each synset are shown.

Synset	Gloss
Direct flow	
be#v#1	have the quality of being; (copula, used with an adjective or a predicate noun)
capable#a#1	(usually followed by ‘of’) having capacity or ability
quality#n#1	an essential and distinguishing attribute of something or someone
have#v#1 hold#v#6 have_got#v#1	have or possess, either in a concrete or an abstract sense
not#r#1	negation of a word or group of words
fashion#n#1 way#n#1 style#n#2	how something is done or how it happens
good#a#1	having desirable or positive qualities especially those suitable for a thing specified
characterize#v#2 characterise#v#1	be characteristic of
virtuous#a#1 just#a#4 good#a#8	of moral excellence
golden#a#4 favored#a#3 fortunate#a#2	supremely favored or fortunate
Inverse flow	
balmy#a#2 mild#a#3	mild and pleasant
top-flight#a#1 top-hole#a#1 topping#a#1	excellent; best possible
mean#a#4	excellent
good#n#2 goodness#n#2	moral excellence or admirableness
wonderfulness#n#1 admirability#n#1	admirable excellence
spiffing#a#1	excellent or splendid
good#a#7	agreeable or pleasing
worthy#a#5	morally admirable
good#a#3	morally admirable
nice#a#6 gracious#a#4 courteous#a#1	exhibiting courtesy and politeness
Bidirectional flow	
be#v#1	have the quality of being; (copula, used with an adjective or a predicate noun)
virtuous#a#1 just#a#4 good#a#8	of moral excellence
decent#a#1 nice#a#2	socially or conventionally correct; refined or virtuous
golden#a#4 favored#a#3 fortunate#a#2	supremely favored or fortunate
rosy#a#4 fortunate#a#3 hopeful#a#2	presaging good fortune
discriminate#a#2 nice#a#5	noting distinctions with nicety
self-respect#n#1 self-esteem#n#2	the quality of being worthy of esteem or respect
beneficial#a#1 good#a#5	promoting or enhancing well-being
cheerful#a#1	being full of or promoting cheer; having or showing good spirits
admirable#a#1	deserving of the highest esteem or admiration

Table 5.3: The ten top-ranked negative synsets returned by each of the three proposed models. Only the first three lemmas for each synset are shown.

Synset	Gloss
Direct flow	
not#r#1	negation of a word or group of words
be#v#1	have the quality of being; (copula, used with an adjective or a predicate noun)
unfortunate#a#1	not favored by fortune; marked or accompanied by or resulting in ill fortune
badness#n#1 bad#n#1	that which is below standard or expectations as of ethics or decency
incorrect#a#2 inappropriate#a#2 wrong#a#8	not conforming with accepted standards of propriety or taste; undesirable
negative#a#1	characterized by or displaying negation or denial or opposition or resistance…
pathetic#a#1 piteous#a#1 misfortunate#a#1	deserving or inciting pity
spoilt#a#2 spoiled#a#3 bad#a#4	(of foodstuffs) not in an edible or usable condition
calumnious#a#1 denigrating#a#1 libellous#a#1	(used of statements) harmful and often untrue; tending to discredit or malign
damaging#a#2 negative#a#8	designed or tending to discredit, especially without positive or helpful suggestions
Inverse flow	
inauspicious#a#3 ill#a#5 ominous#a#2	presaging ill-fortune
wimpy#a#1 wimpish#a#1	weak and ineffectual
down#r#5	to a lower intensity
down#a#6 depressed#a#1	lower than previously
shoddy#a#1 cheapjack#a#1 tawdry#a#2	cheap and shoddy
ill-fated#a#1 ill-starred#a#1 unlucky#a#2	marked by or promising bad fortune
unfortunate#a#3	unsuitable or regrettable
abject#a#2	most unfortunate or miserable
deplorable#a#1 pitiful#a#2 distressing#a#2	bad; unfortunate
scrimy#a#1	dirty and disgusting
Bidirectional flow	
not#r#1	negation of a word or group of words
be#v#1	have the quality of being; (copula, used with an adjective or a predicate noun)
unfortunate#a#1	not favored by fortune; marked or accompanied by or resulting in ill fortune
badness#n#1 bad#n#1	that which is below standard or expectations as of ethics or decency
incorrect#a#2 inappropriate#a#2 wrong#a#8	not conforming with accepted standards of propriety or taste; undesirable
inauspicious#a#1 unfortunate#a#2	not auspicious; boding ill
negative#a#1	characterized by or displaying negation or denial or opposition or resistance…
spoilt#a#2 spoiled#a#3 bad#a#4	(of foodstuffs) not in an edible or usable condition
pathetic#a#1 piteous#a#1 misfortunate#a#1	deserving or inciting pity
calumnious#a#1 denigrating#a#1 libellous#a#1	(used of statements) harmful and often untrue; tending to discredit or malign

model, and this pushes them up in the ranking[7].

This phenomenon does not appear in the inverse flow model. For example, the synsets that appear in the glosses of verbs such as "to be" are unlikely to be ORP-loaded; such verbs thus obtain a low score. The inverse flow model top-ranks those glosses which are almost exclusively composed of semantically oriented terms. Again, the bidirectional flow model trades off between the other two models, producing a ranking which appears to mix the characteristics of the other two.

5.6.2 Stability with respect to the values of α

We have also studied how stable the three models are with respect to variations in the value of the α parameter. Figure 5.2 plots the τ_p measure obtained on the test set as a function of α; all values from 0.00 to 1.00, with 0.01 step increments, have been tested. The figure clearly shows that each model has its own characteristic trend.

The direct flow model produces good results for small values of α, but slowly deteriorates as α increases, even performing worse than the baseline for high enough values of α (for $\alpha = 0.86$ for the ranking by positivity, and for $\alpha > 0.28$ for ranking by negativity)[8].

The inverse flow model produces very good results for almost the entire range of values of α, and is stably better than the baseline. Only for values of α close to 1.00 the inverse flow model produces bad results; however, this is caused by the fact that for this model and high values of α (> 0.90) PageRank turns out to converge very slowly, and is thus stopped at the 1000 iteration limit before the output vector contains stable values.

The bidirectional flow model shows an unexpected trend. For low values of α it produces relatively good results but, as α increases, the model degrades rapidly, producing the worst results.

This analysis confirms the good qualities of the inverse flow model; we now know that it is not only the best performing model after parameter optimization, but that it is also the one that delivers more stable

[7] In order to solve this problem we have also tested a version of the direct flow model in which synset s_k receives the *average*, and not the sum, of the contributions of the synsets s_i such that $s_i \blacktriangleright s_k$; however, this has produced inferior results with respect to the standard direct flow model.

[8] Note that for $\alpha = 0$ the ranking produced by PageRank coincides with that embodied in the e vector before the application of PageRank, i.e., with our baseline.

performance with respect to the chosen value of α.

5.7 SentiWordNet 2.0

The numeric values assigned by PageRank to a synset s_i, which determine its rank position $R(s_i)$, cannot be directly used to define a SentiWordNet-like triplet of positivity, negativity and objectivity values. The **a** vector of PageRank scores is normalized so that the sum of all of its values is one. Given that **a** consists of 115,424 elements, every a_i value is very small. For example, the top-ranked synset for the best positivity result has an a_i value equal to $0.885 * 10^{-5}$. The simple heuristics of taking, for a synset s_i, the two a_i values obtained from the the two distinct best rankings of positivity and negativity, i.e. $a_i^{Positive}$ and $a_i^{Negative}$, and assigning

$$\begin{array}{rcl}\hat{\Phi}(s_i, Positive) & = & a_i^{Positive} \\ \hat{\Phi}(s_i, Negative) & = & a_i^{Negative} \\ \hat{\Phi}(s_i, Objective) & = & 1 - (\hat{\Phi}(s_i, Positive) + \hat{\Phi}(s_i, Negative))\end{array} \quad (5.8)$$

produces almost equally valued triplets for all synsets, with very high objectivity scores, which, evaluated with the same methodology of SentiWordNet 1.0 (see Chapter 4, Section 4.4), have resulted in poor results, almost identical to the baseline[9].

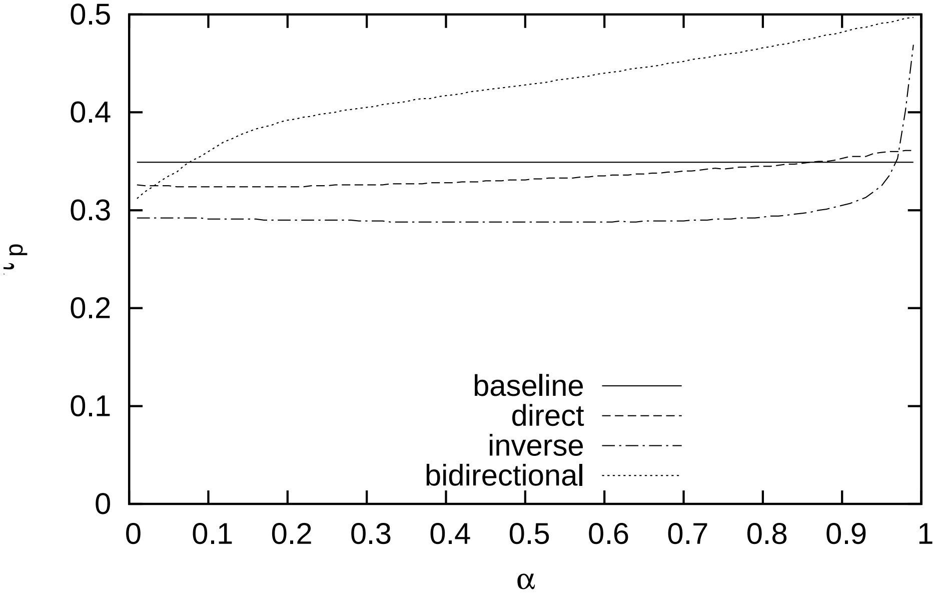
We have found another simple, yet effective, way to build a new version of SentiWordNet. It is based on considering the execution of our PageRank-based algorithm as a *reranking* process:

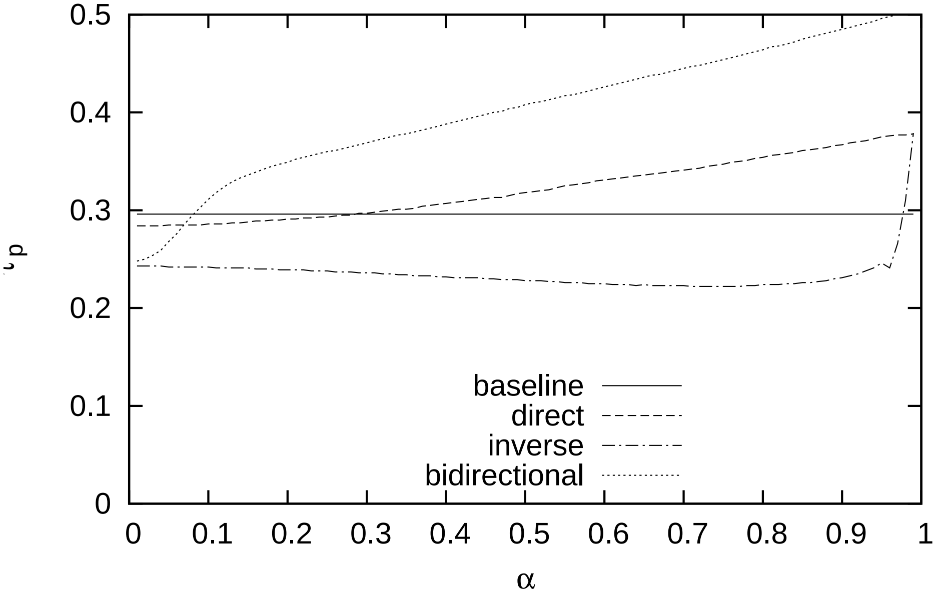
- The configuration that has produced the best results in our ranking experiments, both for positivity and negativity, is the one that uses the "inverse flow" model and takes in input, as the **e** vector, the rankings induced by the SentiWordNet 1.0 values.

- Each of these **e** vectors is itself a complete ranking (one by positivity and one by negativity) of all the WordNet synsets.

[9] We have also experimented on taking a *logarithm*-based function of scores, obtaining again poor results.

Figure 5.2: Plot of the τ_p measure as a function of α using version **e**4 of the **e** vector. The upper and lower figures are about the rankings by positivity and negativity, respectively.





- The **output ranking**, produced by our method, can be thus viewed as a *reranking* of the **input ranking** defined in the **e** vector, i.e. the $\prec$ relation among synsets defined by values in the **e** vector is “corrected” by PageRank using the information contained in the synset graph we built from glosses.

From these considerations we have defined a new SentiWordNet version (2.0), by adopting the following *remapping* function:

$$\begin{aligned}\hat{\Phi}_{out}(s_i, Positive) &= \hat{\Phi}_{in}(S_{in}^{Positive}(R_{out}^{Positive}(s_i)), Positive)\\ \hat{\Phi}_{out}(s_i, Negative) &= \hat{\Phi}_{in}(S_{in}^{Negative}(R_{out}^{Negative}(s_i)), Negative)\\ \hat{\Phi}_{out}(s_i, Objective) &= 1-(\hat{\Phi}_{out}(s_i, Positive)+\hat{\Phi}_{out}(s_i, Negative))\end{aligned} \tag{5.9}$$

where $\hat{\Phi}_{out}$ is the function that defines SentiWordNet 2.0 values, $\hat{\Phi}_{in}$ is the function that defines the **e** vector (SentiWordNet 1.0 in the specific case), R_{out}^p is a function that given a synset s_i returns its position in the output ranking for the property p, S_{in}^p is a function that given a position in the input ranking, for the property p, returns the synset in that position. Note that we can use the output R_{out}^p as input of S_{in}^p because the two rankings have the same number of elements.

In practice, in SentiWordNet 2.0 we have assigned to a synset s_i with a given rank position $R_{out}^p(s_i)$ in the output ranking for p, the same SentiWordNet 1.0 score of the synsets s_j which ranked in the same position in the input ranking for p, i.e. $R_{out}^p(s_i) = R_{in}^p(s_j)$. Then we have assigned it the objectivity score as the difference from one[10].

5.7.1 Evaluation of SentiWordNet 2.0

SentiWordNet 2.0 has been evaluated using the same gold standard and evaluation measure (MSE_μ and MSE_M) of SentiWordNet 1.0, which is described in detail in Chapter 4, Section 4.4. Tables 5.4 and 5.5 report the results of the evaluation, showing that SentiWordNet 2.0 improves over version 1.0 by 9.67%, in terms of reduction of the MSE_M on the whole Micro-WNOp gold standard.

[10] Note that only for 13 synsets, out of 115,424, we have hit the special case $\Phi_{out}(s_i, Positive) + \Phi_{out}(s_i, Negative) > 1$. Only for those cases we have just normalized the two scores to one.

	Micro-WNOp	
	MSE_M	MSE_μ
$\hat{\Phi}_{Positive}$	.532	.630
$\hat{\Phi}_{Negative}$	.512	.618
$\hat{\Phi}_{Objective}$	.462	.311
SentiWordNet 1.0	*.261*	*.182*
SentiWordNet 2.0	**.235** (-9.67%)	**.165** (-9.57%)

Table 5.4: MSE_M and MSE_μ values of SentiWordNet 1.0 and 2.0, and the three baseline functions $\hat{\Phi}_b$, on the whole Micro-WNOp gold standard.

	Micro-WNOp(1)		Micro-WNOp(2)		Micro-WNOp(3)	
	MSE_M	MSE_μ	MSE_M	MSE_μ	MSE_M	MSE_μ
$\hat{\Phi}_{Positive}$	.536	.672	.458	.552	.590	.698
$\hat{\Phi}_{Negative}$	.545	.647	.487	.593	.538	.637
$\hat{\Phi}_{Objective}$	.457	.284	.393	.280	.521	.348
SWN 1.0	*.312*	*.202*	*.195*	*.144*	*.315*	*.216*
SWN 2.0	**.275**	**.178**	**.174**	**.128**	**.289**	**.198**
Δ	-12.01%	-11.65%	-10.86%	-10.71%	-8.33%	-8.39%

Table 5.5: MSE_M and MSE_μ values of SentiWordNet 1.0 and 2.0, and the three baseline functions $\hat{\Phi}_b$, on the three components of the Micro-WNOp gold standard.

This interesting result thus indicates SentiWordNet 2.0 as a better resource with respect to version 1.0. In Chapter 6, we compare these two resources by testing them “in the field”, i.e. using them in an opinion extraction task.

5.8 Conclusions

We have presented three novel random-walk models for ranking WordNet synsets according to how strongly they possess a given ORP; the differences between the three proposed models lies not only in the (obviously different) incidence matrix, but also in the different equations that determine the “hydraulics” of ORP flow. However, by exploiting the properties of the row-normalized incidence matrix of the inverse flow model, all the three models can be recast in terms of the application of PageRank to different matrices.

We have presented comparative results that show, both in a quantitative and qualitative way, the superiority of the inverse flow model. We can thus confidently assert that ORPs may best be seen as flowing from *definiens* to *definiendum*.

We have applied and discussed our models in the context of *opinion-related* properties of synsets. However, we conjecture that these models can be of more general use, i.e., for the determination of other semantic properties of term senses, such as membership in a domain [63].

From the best results in our experiments we have produced an improved version of SentiWordNet. SentiWordNet 2.0 has been defined using a random-walk algorithm that uses the *definiens-definiendum* relation (implicitly) defined by WordNet glosses as the input graph. SentiWordNet 1.0 has been developed using a method based on glosses classification. SentiWordNet 1.0 is given in input to the process that generates SentiWordNet 2.0. The relevant improvement of SentiWordNet 2.0 with respect to SentiWordNet 1.0 shows how much valuable information is contained in glosses.

In the future we plan to re-apply the same algorithms to the forthcoming manually sense-disambiguated version of WordNet. This will allow to eliminate the effect of the noise introduced in eXtended-WordNet by the automatic sense disambiguation phase, thus testing whether the results of this chapter are valid also when “correct”, manually disambiguated glosses are used.

Chapter 6

Automatic Opinion Extraction from Text

Abstract

In this chapter we face an *Opinion Extraction* (OE) task, i.e., identifying in text each expression of subjectivity, the subject expressing it, and the possible target.

We especially focus on how the lexical resources presented in previous chapters could be used in order to improve the performance of an information extraction system on the OE task. We report results on two manually annotated corpora, one in English and one in Italian.

We evaluate our results using some typical evaluation measures for the task, and also using two new evaluation measures we propose, which we contend are better able to capture all the aspects that determine the effectiveness of an information extraction system.

Results show that using features based on our SentiWordNet resource produces a significant improvement with respect to a baseline system not using them and also with respect to the use of features based on other manually-built OM resources. This indicates that the wide coverage of the language guaranteed by SentiWordNet can compensate the errors it contains given its automatic generation.

6.1 Introduction

One of the emerging tasks in OM is *Opinion Extraction* (OE), the task of detecting, *within* a sentence or document, the expressions denoting the statement of an opinion, and detecting therein the sub-expressions denoting the key components and properties (e.g., the opinion holder, the object of the opinion, the type of opinion, the strength of the opinion, etc.) [11, 17, 18, 55, 56]. OE is thus a specialization of *Information Extraction* (IE) to a *non-topical text analysis* task, i.e., analysis of opinion expressions.

A typical IE task consists in extracting from the unstructured information contained in a document some structured information, e.g. extracting the speaker name, the location, the beginning and the end time, from email announcements of seminars [41]. Typical IE subtasks are the recognition of *named entities*[1] (e.g., persons, organizations, locations, temporal expressions), and resolution of coreference (i.e., the recognition of all the expressions that refer to the same object). On such tasks, state-of-the-art IE systems perform with near-human performance, as reported, for example, in the evaluations that have been performed in the Message Understanding Conferences (MUC) [16] organized by DARPA[2].

The OE task seems to be a harder task than typical IE ones, basically because opinions, and subjectivity in general, can be expressed in many different ways. Thus, it is harder to give a rigid structure to an OE problem than a typical IE one. IE tasks can be usually defined as *field-filling* problems, in which a structure with various fields has to be filled (possibly with some fields being optional). For example in the MUC-7 [16] *Scenario Template* IE task, participants were requested to fill a structure with relevant information about air vehicle launch missions (e.g., vehicle information, payload information, the date and location of the launch, and mission information) extracting them from air vehicle launch reports.

For opinion expressions it is not obvious how to give them a structured form, given the many ways subjectivity can be expressed. For example, each of the following sentences express similar opinions, but

[1] The expression "named entity" is used to indicate any entity for which a *rigid designator* exists. A rigid designator, as defined by Saul Kripke, is a name which extensionally identifies an object in every *possible world*. For example, "Bill Clinton" is a rigid designator, "The President of the USA" is not.

[2] `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html`

each one differs from the others in the way the opinion is expressed:

1. `X is good.`
2. `Y scares me.`
3. `X is better than Y.`
4. `I believe X is faster than Y.`

Sentence 1 *explicitly* assigns a positive evaluation to the object X, using the adjective `good`. Sentence 2 does not assign any properties to Y, but reports the (negative) emotive effect of Y on the speaking subject. Thus, sentence 2 can be considered an *implicit* expression of a negative opinion on Y. Sentences 3 and 4 report a comparison between X and Y, but subjectivity is expressed in different ways in them. In sentence 3 subjectivity is expressed by the use of the subjective comparative adjective `better`, with a positive evaluation of X. Sentence 4 is to be considered subjective not because of the use of the comparative adjective `faster` (the property of an object to be `faster` than another is, in principle, an objective and measurable fact), but because of the introduction of the comparison by the verb `believe` which states that the comparison is not the result of an objective measure, but a personal belief of the speaker. It is relevant to note that in this last case the comparison between X and Y can be the expression of a positive appreciation toward X, or Y, depending on the context the comparison is expressed (e.g. a faster CPU is better, a faster-spreading virus is not).

In the OE experiments we present in this chapter we will work on the OE task defined by Wiebe et al. [95, 97] in their works on annotating subjective expressions in language. These works focus on *annotating*[3] in text, either manually or automatically, the *expressions of private state* (EPSs). A private state is "an internal state that cannot be directly observed by others", and as such includes "opinions, beliefs, thoughts, feelings, emotions, goals, evaluations, and judgments" [97, pp. 168]. Since opinion and emotion are arguably the two most important dimensions of private states, we will sometimes call (consistently with [97]) EPSs *expressions of opinion and emotion.* One of the results of their work is the

[3] In our OE task, we will use the verbs *to annotate* and *to extract* as synonyms, with the intended meaning of recognizing the relevant textual expressions in a document, with respect to EPSs (see Section 6.5 for a detailed definition of the task).

MPQA corpus, a corpus of 535 documents, in English, in which EPSs have been manually annotated, along with other relevant information, as detailed in Section 6.3.2.

The goal of our experiments is to make a comparative evaluation of the effect of the use of the OM lexical resources presented in previous chapters in a real OM application. We use, as our OE system, YamCha [58], a well-performing IE system based on SVMs, and test the impact on performance of the various lexical resources. Even though our OE system does not obtain state-of-the-art performance, we show that the use of SentiWordNet could produce a significant increment in effectiveness with respect to the baseline system and, more importantly with respect to the other lexical resources currently available to the scientific community.

We run our experiments on two benchmarks: the MPQA corpus, and I-CAB Opinion [35], a corpus of Italian news manually annotated by EPSs, using the same annotation language of the MPQA corpus. The experiments on the I-CAB Italian corpus allow us to show the possibility of an effective cross-language use of SentiWordNet.

For the evaluation of our experiments we use some typical evaluation measures used in IE and OE. These measures are based on a model that considers each annotated textual expression as a single entity. We show that this model has some limitations and also produces some undesirable effects on the evaluation measures. We also propose two new evaluation measures, based on viewing each *token* composing the text (i.e. any distinct alphanumeric expression, separated from the others by blanks and punctuation) as an entity to be *labeled* as belonging or not to an annotation. We contend that these measures allow to perform a more rigorous evaluation of experiments, capturing all the aspects that determine the effectiveness of an IE system.

6.1.1 Chapter outline

In Section 6.2 we review in some detail the related literature on OE. Section 6.3 goes into the details of the OE task we face, describing the annotation language and the two benchmark corpora. In Section 6.4 we discuss the evaluation measures used to evaluate our experiments. Section 6.5 presents the OE system we used. Section 6.6 gives details on all the experiments and Section 6.7 reports their results. Section 6.8 concludes.

6.2 Related work

6.2.1 Bethard et al.

Bethard et al. [9] have focused on the task of identifying *opinion propositions* in text. In their definition an opinion proposition is a propositional verb argument that expresses an opinion. Their goal is to identify such opinion propositions in sentences, rather than marking the entire sentence as opinionated. This is, in fact, a first step into performing OE from text. For example, in the sentence "`From the first time I met you I believe you are mad`", it is the proposition "`you are mad`" that actually expresses the opinion.

The authors propose to identify opinion propositions in a sentence by classifying the constituent of the parse tree of the sentence. They have evaluated two systems, both based on SVM algorithms: a one-tier system that directly classifies the parse tree elements as opinion propositions or not, and a two-tier system which first classifies the parse tree elements as propositions or not, and then classifies propositions as carrying opinions or not. They test their system on a corpus of 5,139 sentences obtaining a 58.02% precision and a 51.37% recall with the one-tier system, and a 67.97% precision and a 43.72% recall with the two-tier system.

6.2.2 Wiebe et al.

Wiebe et al. [97] is currently the most relevant work on the annotation of opinions in text. The main focus of their work is on the definition of an annotation language able to capture the various expressions of subjectivity in text.

They propose an opinion markup language, which we describe in detail in Section 6.3.1, and which is used to annotate in text the expressions of opinion along with the *opinion holder*, i.e., the subject expressing the opinion and the possible *opinion target*, toward which the opinion is expressed. They have used this language to annotate a corpus of English news, the MPQA corpus[4], which as been used in many OE works (the most relevant ones are described in the following related works sections). Their work also presents a rich study on inter-annotator agreement, which we describe in Section 6.7.2, when discussing the results of

[4] `http://www.cs.pitt.edu/mpqa/`

our experiments.

The I-CAB Opinion corpus (see Section 6.3.3), used in our OE experiments on the Italian language, has been developed using the same annotation language and annotation tools of the MPQA corpus.

6.2.3 Kim and Hovy

Kim and Hovy [55] have worked on the task of recognizing the *opinion holder*, i.e. the holder of an opinion expression. Their work is focused on recognizing opinion holders for the use in a question answering system. Given in input a question like "`What does X think about Y?`", the recognition of the opinion holder allows to eliminate from the candidate answers all the opinions about Y which do not come from X.

Given a sentence containing an opinion expression E, their system identifies many opinion holder candidates H, looking for entities and noun phrases in the sentence. For each pair $\langle E, H \rangle$ the system builds a representation, using features extracted with sentence parsing tools. The representations are provided to a *Maximum Entropy* algorithm, previously trained on a training set of manually labeled $\langle E, H \rangle$ pairs, that ranks the candidates from the most probable to the less probable one. The system has been evaluated on data extracted from the MPQA corpus, in which 863 $\langle E, H \rangle$ pairs have been used for training and 98 for testing. The evaluation measure used is the accuracy in returning the correct opinion holder for a given opinion in the top position of the rank. The system obtains a 62% accuracy over a 39% baseline.

6.2.4 Choi et al., 2005

Choi et al. [18] have worked of the identification of opinion holders too. They model the task as an Information Extraction problem, in which each token composing a sentence has to be classified as belonging or not to the expression identifying an opinion holder.

Their system is based on the use of *Conditional Random Fields* [59] (CRFs) algorithms. The construction of token representations is based on a rich set of syntactic features, subjectivity features extracted from various OM lexical resources, and also on *extraction patterns*. Such extraction patterns are automatically generated using the AutoSlog system [79].

They have tested their system on the MPQA corpus, measuring the effectiveness in recognizing the AGENT tag (see Section 6.3.1), which identifies the opinion holder in text. We compare their results with ours, in Section 6.7.

6.2.5 Choi et al., 2006

In a following work, Choi et al. [17] have investigated the effects of performing a *joint* extraction of opinion holders and opinion expressions.

They use a global inference approach in which entities involved in opinion expressions (i.e. the opinion holder and the opinion itself) are extracted separately by means of a method similar to [18], but designed to have a higher recall. Then a global inference procedure, implemented using *integer linear programming*, is applied in order to produce the best pairing of opinion holders and opinion expressions, by exploiting their mutual dependencies and relations.

They have tested their system on the MPQA corpus, similarly to [18]. We compare some of their results with ours, in Section 6.7.

6.2.6 Breck et al.

Breck et al. [11] is currently the most complete study on performing opinion extraction on the MPQA corpus.

Similarly to Choi et al. [18], their system is based on CRFs and extraction patterns. A very large number of features are used to define token representations, using various lexical resources. They have tested the system on the recognition of the various annotation elements of the annotation language defined in [97]. In particular, they have tested the system with various subsets of features, defined by selecting only some of the lexical resources, in order to identify which one of them gives the most relevant contribution.

Again, given the use of a standard experimental setting, we compare their results with ours, in Section 6.7.

6.3 Annotating expressions of opinion and emotion in text

In order to define our OE task, we have followed the work of Wiebe et al. [97] on annotating subjective expressions in language.

We have adopted (what we here call) the WWC markup language developed in their work, since it was the result of the arguably most important annotation effort (the one which led to the development of the MPQA corpus) in the opinion extraction literature. In this section we present a brief introduction to WWC, referring the reader to [97] for a more detailed description.

6.3.1 The WWC markup language

The WWC markup language provides five types of *tags* (here indicated in SMALL CAPS), to identify the various components involved in EPSs. Each such tag can be further qualified by means of a number of *attributes* (here indicated in `typewriter` font). Aside from specifying in more detail the role played by the real-world entities denoted by the tagged expressions, attributes also allows to establish relations among the entities that play different roles in the same EPS.

In WWC every EPS is mapped into a *private state frame*, i.e., a structured object in which the real-world entities that play a role in the EPS are annotated by means of the tags and further qualified by means of the attributes (see Table 6.1 for an example). In each private state a *source agent* holds a private state, optionally toward a *target agent*. WWC identifies three kinds of private states:

1. the explicit mention of a private state (e.g., "`I fear the Greeks, even when they bring presents`");

2. a speech event expressing a private state (e.g., "`You said you love her.`");

3. an expressive subjective element (e.g., "`He is a nice person`").

WWC also allows annotating *nested* EPSs in which the target agent is itself a private state (e.g., "`John wrote me that Mary said I love pizza`"); the structured nature of private state frames naturally allows expressions at arbitrary levels of nesting to be represented.

A textual expression (*text span*, in WWC terminology) identifying the source agent or the target agent of a private state is annotated with the AGENT tag, which assigns a unique (at the document level) identifier to the entity denoted by the expression. Since EPSs can be nested, it is natural to identify the outermost source of every EPS in a given text as the author of the text itself; by convention, the identifier denoting the author of the text is "writer".

The explicit mention of a private state (Type 1 above), or a speech event expressing a private state (Type 2 above) are annotated using the DIRECT-SUBJECTIVE tag. The text span expressing either the mention of the private state or the speech event is identified, and the following attributes are specified:

- `intensity`: the intensity of the private state (`low` to `extreme`);
- `expression − intensity`: the contribution of the speech event expression to the intensity of the private state, e.g. "`say`" vs. "`cry`" (`neutral` to `extreme`);
- `insubstantial`: a Boolean flag indicating whether the private state is substantial to the discourse or not (e.g., hypothetical clauses are not substantial);
- `polarity`: the attitude of the private state, ranging on the values `positive`, `negative`, `other` and `none`;
- `source`: the `chain` of agents expressing the private state;
- `target`: (optional) the agent which is the target of the private state.

The use of chains of agents to identify targets is the key WWC device for the expression of nested private state frames. For example, in the sentence "`John wrote me that Mary said I love pizza`" the DIRECT-SUBJECTIVE annotation related to the verb "`said`" has the target attribute equal to "writer/john/mary", because it is the author of the text who reports that John wrote that Mary said something about a private state.

Reported speech about objective facts is also annotated (e.g., "`John said he is 30`"), using the OBJECTIVE-SPEECH-EVENT tag. A source agent and a target agent are assigned to the annotated text[5].

WWC also includes an INSIDE tag, used for identifying the scope of a speech event (e.g., "`Mary said I love pizza`"). This tag has not been

[5] In the version of MPQA used in this work (1.2) the target agent attribute, although defined in WWC, has not been used in the manual annotation of documents.

used in MPQA (except for automatically marking an entire sentence as an INSIDE for "writer").

Finally, subjective expressions in text are annotated using the EXPRESSIVE-SUBJECTIVITY tag, that identifies the text span of a subjective expression and qualifies it by means of three attributes: source agents chain, intensity, and polarity of the expression.

Table 6.1 illustrates the private state frame generated by the sentence "`John wrote me that Mary said I love pizza`".

6.3.2 MPQA

The WWC annotation language has been used in [97] to manually annotate EPSs in the MPQA corpus[6]. MPQA stands for Multiple Perspectives in Question Answering. One of the intended use of the MPQA corpus is to investigate the possibility of building a question answering system that takes into account the perspective of the information source. For example, the answer to the question "Was the 2006 election in Italy fair?" could receive different answers from different (politically oriented) sources. The MPQA corpus consists of 535 documents (10,657 sentences), which are English versions of news articles collected from 187 press sources around the world. News date from June 2001 to May 2002.

In [97] an inter-annotator agreement (IAA) study is reported. IAA has been measured on a set of 13 documents (210 sentences), among three annotators. The results of the IAA study are discussed in Section 6.7.2

Many OE works [9, 11, 18, 17, 55, 56, 97] have used the MPQA corpus for their experiments, and in most of these works it has been subdivided into two parts: a validation set, consisting of the first 135 documents, used for optimizations, and a test set, composed by the remaining 400 documents (8,297 sentences), on which final experiments are executed by performing a 10-fold cross validation. We have adopted this subdivision in our experiments, in order to be able to compare our results with some of the results already published (see Section 6.7).

[6] `http://www.cs.pitt.edu/mpqa/databaserelease`

```
AGENT (text: “John”;
          id: john);
AGENT (text: “Mary”;
          id: mary);
AGENT (text: “I”;
          id: andrea);
AGENT (text: “pizza”;
          id: pizza);
DIRECT-SUBJECTIVE (text: “wrote”;
          intensity: low
          expression − intensity: neutral;
          polarity: positive;
          insubstantial: false;
          source: writer/john;
          target: mary);
INSIDE (text: “Mary said I love pizza”;
          source: writer/john);
DIRECT-SUBJECTIVE (text: “said”;
          intensity: low;
          expression − intensity: neutral;
          polarity: positive;
          insubstantial: false;
          source: writer/john/mary;
          target: andrea);
INSIDE (text: “I love pizza”;
          source: write/john/mary);
DIRECT-SUBJECTIVE (text: “love”;
          intensity: high;
          expression − intensity: high;
          polarity: positive;
          insubstantial: false;
          source: writer/john/mary/andrea;
          target: pizza)
```

Table 6.1: The private state frame generated by the sentence “`John wrote me that Mary said I love pizza`”.

6.3.3 I-CAB Opinion

I-CAB Opinion [35] is the results of annotating the Italian Content Annotation Bank (I-CAB) [64] by EPSs, using the WWC markup language.

The Italian Content Annotation Bank (I-CAB) [64] is a corpus of newspaper articles in the Italian language, manually annotated with semantic information of various types, including TEMPORAL EXPRESSIONS, different types of entities (such as PERSON ENTITIES, ORGANIZATION ENTITIES, LOCATIONS, and GEO-POLITICAL ENTITIES), and RELATIONS between such entities (such as, e.g., "affiliation", relating a person to the organization he/she is affiliated to)[7].

I-CAB has been developed with the aim of making it both (a) a standard resource for supporting the development of algorithms for the automatic extraction of different types of information, and (b) a benchmark for testing such algorithms. Indeed, I-CAB has served as the reference resource and benchmark within several tracks of EVALITA'07, a campaign for the evaluation of NLP tools for the Italian language [14].

I-CAB consists of 525 articles published by L'Adige[8], an Italian local newspaper, on four different days (September 7, September 8, October 7, October 8, all in 2004). The articles are from the Current Events (87 articles), Cultural News (72), Economic News (54), Sports News (123), and Local News (189) sections of the (print edition of the) newspaper, and are subdivided into a training set of 335 articles (with an average length of about 339 word tokens) and a test set of 190 articles (with an average length of about 363 word tokens).

The choice of using the WWC markup language for annotating I-CAB, thus producing I-CAB Opinion, has been made for many reasons:

- to avoid "reinventing the wheel" and instead to leverage on past experience from other researchers;
- to ease comparisons between the same linguistic phenomena as occurring in different languages.

This is completely in keeping with the policy adopted by Magnini et al. [64] for annotating I-CAB along the other dimensions described above, given that [64] adopted markup languages previously developed within the ACE program[9].

[7] `http://tcc.itc.it/projects/ontotext/icab.html`

[8] `http://www.ladige.it/`

[9] `http://www.nist.gov/speech/tests/ace/`

For annotating I-CAB by EPSs the (freely available) GATE[10] tool developed at the University of Sheffield [21] has been used. It is the system originally used in [97], and it already includes the tools to support the manual annotation of document using the WWC markup language.

Consistently with the other types of annotation on I-CAB described in [64], our EPSs annotations are encoded in MEAF [8], an XML-based format compliant with the guidelines set by the Text Encoding Initiative (TEI). However, since the annotations generated by GATE are not in MEAF, we had to implement a translator from the format generated by GATE into MEAF. One of the advantages of having the various types of annotations expressed in the same format is that it allows us to interlink them and, navigating across the various types, to discover new relevant information. For example, connecting the AGENT annotations, with the name entities annotations, and then using the coreference information about entities, enables us to find all the expressions of opinion in which a given entity has some role.

6.4 Evaluation measures

In order to evaluate our experiments we use two different kinds of measures:

- A first class of measures, typically used in IE and OE literature, is based on considering each annotated text span as a single entity; the evaluation is based on comparing the matches among the two sets of *correct* annotations, from the benchmark corpus, and *predicted* annotations, from the OE system.

- The other class of measures we propose in this work for the first time, is based on considering each *token* (i.e., any distinct alphanumeric expression, separated from the others by blanks and punctuation) composing the text as a single entity to be classified as belonging or not to an annotation (for each possible annotation type). The evaluation is done by measuring the classification quality by using standard IR evaluation measures, such as precision, recall and F_1.

[10] http://gate.ac.uk/

We contend that the annotation-based measures have some relevant limitations, which we point out in Section 6.4.1, and that the token-based measures allow to perform a more rigorous evaluation of experiments, capturing all the aspects that determine the effectiveness of an IE system, as we detail in Section 6.4.2.

6.4.1 Annotation-based measures

Annotation-based measures consider each annotated text span as a single entity. The evaluation is thus made by comparing the matches among two sets: the one of *correct* annotations $G_X = \{g_1, \ldots, g_n\}$, and the other of the *predicted* annotations, $P_X = \{p_1, \ldots, p_m\}$, for various possible types of annotation tag X[11]. A first point to note is that P may contain an arbitrary number of elements, which is not forced to be equal to the number of elements contained in G. Moreover, the annotations in P may obviously refer to any portion of text in the annotated documents, without any relation with G. A consequence of this facts is that it is impossible to establish any one-to-one relation between the elements in G and P.

The typical IE approach [49, 60] is to define a predicate $match(g, p) \rightarrow \{True, False\}$, which determines what is a match between two elements $g \in G$ and $p \in P$, and to then use this predicate to compute an approximate version of the evaluation measures of precision (π), recall (ρ), and F_1 [62]:

$$\pi(G,P) = \frac{|\{p|p \in P \land \exists g \in G : match(g,p)\}|}{|P|} \tag{6.1}$$

$$\rho(G,P) = \frac{|\{g|g \in G \land \exists p \in P : match(g,p)\}|}{|G|} \tag{6.2}$$

$$F_1(G,P) = \frac{2\pi(G,P)\rho(G,P)}{\pi(G,P) + \rho(G,P)} \tag{6.3}$$

Three widely adopted definitions [11, 41, 60] for the *match* predicate are:

[11] The possible values in our case, according to the WWC markup language, are: AG = AGENT, DS = DIRECT-SUBJECTIVE, ES = EXPRESSIVE-SUBJECTIVITY, and OSE = OBJECTIVE-SPEECH-EVENT.

overlap: $match_{overlap}(g, p) = True$ iff the two annotations have *any overlap* in text;

head: $match_{head}(g, p) = True$ iff the two annotations *start* from the same position in text;

exact: $match_{exact}(g, p) = True$ iff the two annotations *start and end* at the same positions in text;

For example, given the following gold standard annotation and three system annotations:

$g =$ “`he is a` $[_{ES}$`good and honest`$]_{ES}$ `boy`”[12]

$p_1 =$ “`he is a good and` $[_{ES}$`honest boy`$]_{ES}$”

$p_2 =$ “`he is a` $[_{ES}$`good and`$]_{ES}$ `honest boy`”

$p_3 =$ “`he is a` $[_{ES}$`good and honest`$]_{ES}$ `boy`”

the $match_{overlap}(g, p_i)$ predicate is true for all annotations p_i, the $match_{head}(g, p_i)$ predicate is true for annotations p_2 and p_3, while the $match_{exact}(g, p_i)$ predicate is true only for annotation p_3.

Unfortunately, these predicates have some drawbacks. The overlap match predicate tends to overestimate the performance of a system that tends to produce long annotations. A “cheating” system that annotates the whole document in a single annotation obtains a perfect performance according to this predicate. Works using these measures usually report the average number of overlapping annotations in P (or G) for each annotation in G (resp. P), alleging that a value near one (average overlap is 1.08 in [17] and 1.13 in [11]) show a “reasonably behaving” system. However, it is not possible to define a threshold value to determine when a system is cheating or not, or just to decide what is the influence of the expected average overlap on performance. For example, given the system X with $F_1 = 0.8$ and $overlap_{avg} = 1.1$ and system Y with $F_1 = 0.85$ and $overlap_{avg} = 1.3$, which is the best one? In Section 6.7.3 we report a study on the relation of the $overlap_{avg}$ value with the performance measured by the overlap measure.

[12] In the following we will use the square brackets $[_X$ and $]_X$ to identify the boundaries of an annotation.

On the opposite side, the head match and (even worse) the exact match predicates, are too strict in their evaluation, especially for the OE tasks, because they treat all the approximate matches as full errors. For example, including the leading article in the annotation g, presented above, "`he is` $[_{ES}$`a good and honest`$]_{ES}$ `boy`", will be counted by both predicates as an error, despite the minor difference.

All these match predicates do not take into account the *degree* of overlap between two compared annotations. For instance, two annotations that are each 10 words long and that overlap by 1 word only receive the same partial credit as two annotations that are each 10 words long and that overlap by 9 words, which is unintuitive.

Last, and may be least, it is not possible to compute a full *contingency table* by using annotation-based measures, due to the lack of the concept of *true negatives.* In fact it is possible separate annotations in three categories:

- *true positives*: annotations in G that have a match with some annotations in P[13];
- *false negatives*: annotations in G that do not have a match with any annotations in P;
- *false positives*: annotations in P that do not have a match with any annotations in G;

but the unannotated text (which in theory contributes to determine true negatives) it is not considered in the matching process. This hampers the possibility of using evaluation measures which require a full contingency table, such as Cohen's κ [19, 36], which is open used to measure inter-annotator agreement.

6.4.2 Token-based measures

The new class of measures we propose are based on considering each *token* composing the text as a single entity to be classified as belonging or not to an annotation. In practice, we reformulate the problem of evaluating annotations as a problem of evaluating tokens classification,

[13] Note also that the count of true positives is *non-symmetric*, i.e. inverting the role of G and P the computed value may change

in which the various tags are the possible categories that can be assigned to tokens.

More formally, in a *token model* evaluation, given a document d, composed by a sequence $\{t_1, \ldots, t_k\}$ of tokens, we consider a token t_i as belonging to the category c_X, where X is one of the annotation tags, iff there exists at least one annotation $g \in G_X$ which includes the token t_i. The predictions from the system are interpreted similarly, finally obtaining two token classifications which can be compared using any standard evaluation function.

Given the following annotated sentences:

$s_g =$ "`it is a` $[_{ES}$`love hate`$]_{ES}$ `relationship`"

$s_1 =$ "`it is` $[_{ES}$`a love`$]_{ES}$ `hate relationship`"

$s_2 =$ "`it is a` $[_{ES}$`love`$]_{ES}$ $[_{ES}$`hate`$]_{ES}$ `relationship`"

$s_3 =$ "`it is a` $[_{ES}$`love hate`$]_{ES}$ `relationship`"

the token model evaluation considers the sentence s_1 to contain two errors with respect to the gold standard sentence s_g, a false positive for the token `a`, and a false negative for the token `hate`, while the remaining tokens are correctly classified.

In the token model the two sentences s_2 and s_3 are considered to be correctly annotated. The problem here is that s_2 contains an error because, even if the subjective expression has been entirely and correctly annotated, it has been split in two annotations when the two terms should be instead linked together into a single annotation.

The *token & blank model* is a simple extension of the token model which enables to capture also this case. In this model the document d is considered as an alternated sequence of tokens and blanks $\{t_1, b_1, t_2, \ldots, t_{k-1}, b_{k-1}, t_k\}$, where tokens are labeled in the same way as in the token model, and a blank b_i is considered to be labeled as belonging to the category c_X, iff *both* tokens t_i and t_{i+1} are labeled by category c_X. Then the evaluation is performed in the same way as in the token model.

The token & blank model is thus able to spot the difference between s_2 and s_3, assigning in s_2 a false negative classification error to the blank between `love` and `hate`. s_3 is instead evaluated as perfectly annotated.

We deem that token-based models are rather intuitive, given also the fact that it is a common approach for IE system to treat the annotation problem as a token classification problem.

With respect to the annotation-based measures, token-based measures have many advantages:

- Token-based measures are robust with respect to "cheating" systems. For example, the system producing a single document-long annotation, which is evaluated as perfect by the $match_{overlap}$-based measure, is equivalent to a *trivial acceptor* system that assigns the analyzed tag to all tokens. Such system, which is typically used to define the trivial *baseline* for the F_1 measure in classification systems, scores the maximum recall, but a very low precision[14] and thus a low F_1.

- The number of entities under evaluation is constant, regardless of the number of annotations produced by the evaluated system. In practice, an overannotating (or underannotating) system is penalized, due to the fact the many of its annotations will generate false positives (resp. negatives).

- It is possible to compute a full contingency table, thus allowing the evaluation of measures like Cohen's κ.

- Values in the contingency table are tolerant to the role of the two classifications being compared, i.e., switching gold standard and predictions just swap the number of false positives and false negatives, leaving unchanged the number of true positives and true negatives.

- Token-based measures are tolerant to minor errors, such as adding a spurious token to a long annotation.

- At the same time, token-based measures are strict on assigning high performance scores. The perfect performance is returned by the token & blank model measure only when the gold standard and the predicted annotations are exactly the same (like for the exact match).

[14] Equivalent to the ratio of annotated tokens in the gold standard over the total number of tokens.

6.5 The information extraction system

As the learning and classification engine of our OE system we have used YamCha[15] [57, 58]. As the YamCha name itself states[16], the YamCha system is a general purpose system for performing text chunking tasks. As the core learning algorithm YamCha uses SVMs, and it specifically uses the TinySVM package[17]. YamCha scored the best performance in the CoNLL-2000 Chunking Shared Task [84].

The input data for the training phase is composed by a sequence of tokens $\{t_1, \ldots, t_k\}$, where each token t_i is represented by a set of *features* $F_i = \{f_i^1, \ldots, f_i^n\}$ (e.g. the term at position i, its POS and lemmatized version) and a target classification label c_i.

YamCha also provides a useful tool to enrich the features representation of a token by adding information from the tokens in a specified neighborhood *window*. For example, specifying a $[-2, +2]$ window, the representation for the token t_i will be also composed by the features of the two preceding and following tokens, $F_i = \{f_{i-2}^1, \ldots, f_{i-2}^n, \ldots, f_i^1, \ldots, f_i^n, \ldots, f_{i+2}^1, \ldots, f_{i+2}^n\}$, thus allowing the learner to capture information from the context surrounding the observed token. Moreover, in addition to these *static features*, i.e. features that are known before performing the classification, it is possible to add to token representations a window on *dynamic features*, i.e., the labels assigned to neighboring tokens after the classification, so as to use information from previous classification decision to the current one. The dynamic feature window can only point to tokens that precede the currently analyzed one. At training time the values of the dynamic features are extracted from training data, while at classification time they are effectively computed on the fly.

In our experiments we have considered the annotation of each tag in the WWC markup language as a distinct task, thus producing four versions of the data[18] where all the annotations for each tag have been separately converted into the YamCha format, using the IOB2 format. The IOB2 format is a standard format for the representation of text chunking problems. The label “O” is assigned to tokens outside any annotation, the label “B” is assigned to tokens at the beginning of an

[15] `http://www.chasen.org/~taku/software/yamcha/`

[16] The YamCha acronym stands for Yet Another Multipurpose CHunk Annotator.

[17] `http://www.chasen.org/~taku/software/TinySVM/`

[18] We have not investigated on the INSIDE tag, which we consider, at the current time, of minor interest.

Token	Term	POS	Lemma	Label
$t-3$	It	PRP	it	O
$t-2$	is	VBZ	be	O
$t-1$	a	DT	a	O
t	love	NN	love	B-ES
$t+1$	hate	NN	hate	I-ES
$t+2$	relationship	NN	relationship	O
$t+3$	.	PUNC	.	O

Figure 6.1: Example of tokens representations, with features included in the static and dynamic windows of the token t underlined.

annotation, and the label "I" is assigned to tokens in any other position inside an annotation.

Figure 6.1 shows an example of tokens representations, highlighting the set of features used to represent the token t when a $[-2, +2]$ static window and a $[-2, -1]$ dynamic window are specified.

6.6 Experiments

We have defined our experiments with the goal of measuring the impact of using the various OM lexical resources we have presented in previous chapters in an OE tasks. This goal is twofold: evaluate the impact of using OM lexical resources in OE, and "indirectly" compare the performance of the various OM lexical resources, in order to spot which one performs the best when put in use.

We have thus prepared various versions of the two annotated corpora used, each one with some specific information extracted from documents by using the various OM lexical resources.

In details, for the MPQA corpus we have tested five *features sets*:

BASE: in this version each token is represented by the following features:

- the term identifying the token, exactly as it appears in the text;
- the lowercase version of the term;

- a four-valued feature that specifies the capitalization properties of the term { AllLowerCase, AllUpperCase, Mixed, NotWord };
- the part of speech of the term, obtained by using the Brill tagger [12].

GI: this version includes the base features, plus a feature which indicates if the term is labeled as either Positive or Negative in the General Inquirer's lexicon (see Chapter 1, Section 1.4.1). This resulted in tagging 1,416 distinct terms in the MPQA corpus as subjective, for a total of 98,130 occurrences.

HM: this version includes the base features, plus a feature which indicates if the term appears in the subjectivity lexicon defined by Hatzivassiloglou and McKeown [44] (see Chapter 1, Section 1.4.1). This resulted in tagging 747 distinct terms in the MPQA corpus as subjective, for a total of 31,620 occurrences.

SWN1: this version includes the base features, plus a subjectivity feature that indicates if the term is one of the 2,645 distinct terms in the MPQA corpus that has a high subjectivity score in SentiWordNet 1.0 (see Chapter 4), for a total of 171.467 occurrences. We have defined the SentiWordNet subjectivity score for a term as the sum of positivity and negativity scores of all the synsets the term belongs to. When the subjectivity score for a term is higher than 0.5, we consider it as subjective.

SWN2: this version is almost identical to SWN1, with the only difference that the subjectivity scores have been extracted from SentiWordNet 2.0 (see Chapter 5), thus identifying 2,333 subjective terms in the MPQA corpus, for a total of 176,600 occurrences.

.

ALLSUBJ: the union of all the features defined in the previous features sets.

For the I-CAB Opinion corpus, we do not have any Italian language lexical resource dedicated to OM. This is the case in which having a lexical resource aligned to WordNet is a great vantage point. We have in fact used MultiWordNet [78], in order to map SentiWordNet scores to Italian synsets[19], and then computing the subjectivity lexicon for Italian terms. Thus, for the I-CAB Opinion corpus we have tested three features sets:

BASE: in this version each token is represented by the following feature that we have obtained from the other I-CAB levels of annotation:

- the term identifying the token, exactly as it appears in the text;
- the lowercase lemmatized version of the term;
- a four-valued feature that specifies the capitalization properties of the term { AllLowerCase, AllUpperCase, Mixed, NotWord };
- the part of speech of the term;
- for verbs: mood, tense, person and number.

SWN1: base features plus a subjectivity feature based on subjectivity scores extracted from SentiWordNet 1.0 and mapped to the Italian language using MultiWordNet (version 1.4.1). This process resulted in identifying a set of 541 terms appearing in the I-CAB Opinion corpus, for a total of 19,051 occurrences.

SWN2: like SWN1, but based on SentiWordNet 2.0, identifying 523 subjective terms, for a total of 17,610 occurrences.

We have used the windowing option of YamCha specifying a $[+2, -2]$ static window and a $[-2, -1]$ dynamic window. These values have been determined by running a 10-fold cross validation experiment on the validation part of the MPQA corpus, composed by 135 documents. In these experiments we have compared different windows specifications of the form $[+i, -i]$ for static window and $[-i, -1]$ for dynamic window, for $i \in [1, 5]$. We have obtained the best results for $i = 2$, which thus has been selected as final value for the parameter in all the experiments.

[19] We have to admit this has not been an easy task, given that MultiWordNet (we have used the latest version 1.4.2) is currently aligned to WordNet 1.6. So the translation required also to perform a mapping of SentiWordNet scores from WordNet 2.0 to WordNet 1.6.

On the MPQA corpus we have performed a 10-fold cross validation on the testing part of the corpus, composed by 400 documents. On the I-CAB Opinion corpus we have used the default split of the corpus [64], composed by 335 training documents and 190 test documents.

6.7 Results

Results of the OE experiments on the MPQA corpus are reported in Tables 6.2 and 6.3. In the evaluation of all the experiments we have followed the previous literature [11, 18], i.e., we have analyzed all the annotations, or token classifications, together, not averaging partial results on documents.

From a global review of results, a general trend clearly emerges: the use of subjectivity feature sets produces an improvement in global effectiveness, as measured by the various F_1 versions, produced by a high gain in recall, which more htan compensate for a small loss in precision. Thus, we can hypothesize that the subjectivity features sets contribute to the OE process in allowing it to spot more text spans in which relevant information, for the annotation task, appears. At the same time this has the adverse effect of introducing a source of false positives, which fortunately has a minor effect on global performance.

The best performance of the subjectivity feature sets is observed on the EXPRESSIVE-SUBJECTIVITY WWC tag. This is a reasonable result, given the affinity between the semantic of the tag and the information contained in the features sets.

The average improvement, in terms of token-based F_1, over the various tags with respect to the BASE feature set, is 2.23% for the GI features set, 1.35% for HM, 4.40% for SWN1, 4.30% for SWN2, and 5.79% for ALLSUBJ. The SWN1 and SWN2 feature sets always perform better than the GI and HM features sets. In Section 6.7.1 we report the results of statistical significance measurements in order to assess the relevance of these results.

The better performance of the SentiWordNet-based features sets indicates that their wide coverage of the language largely compensates for their innaccuracies, due to their automatic generation. For example, in the SWN1 feature set the term `phone` is erroneously marked as subjective, but this feature set also includes, correctly, the terms `advantageous` and `insulting` which are missing instead from both the GI and the HM feature sets.

It is interesting to note the the ALLSUBJ feature set, which combines all the other feature sets, has always scored the best performance, suggesting that none of the tested feature sets "contains" the others, and each one contains *relevant* information about subjective language that the others do not capture.

Tables 6.2 and 6.3 also report the state-of-the-art results currently known in the literature, obtained on the same task. The comparison clearly indicates that our system is far from these best results. The two main differences between our system and those of [11, 17, 18] are: (a) the learning algorithm (we use SVMs, while all the other systems use *Conditional Random Fields* [59] (CRFs)) and (b) the features used to represent tokens. A further difference with [17] is in the more complex approach used in that work, with a system that performs a joint annotation of the various tags.

With respect to point (a) we can observe, *a-posteriori*, the possibility that CRFs could be better suited for the task. This statement is in part supported by the overall comparison of results, and most important, by the evidence that comes from the "feature ablation" study reported in Breck et al. [11], in which their CRFs system is tested on various feature sets, including a "base" one (see [11, Table 5]) which should be almost equivalent to our "BASE" feature set. Comparing our "BASE" results with Breck's "base" results (reported as *CRFs base* in Table 6.2), the CRFs clearly produce better results compared to SVMs on the overlap measure, and are only slightly worse on the exact measure.

With respect to point (b) we point out that we have designed our experiments with the aim of creating an "isolated" environment for the evaluation of the impact of the various lexical resources on the OE tasks. Thus we have reduced the BASE features to a somewhat minimal definition and we have not used any advanced NLP tool. As also our experiments with the ALLSUBJ feature set show, the use of more lexical resources to define subjectivity features is likely to produce an improvement in performance. The "feature ablation" study of Breck et al. [11] again supports us, showing an almost monotonic increase in performance with the inclusion on new subjectivity features into the token representations.

Results on the I-CAB Opinion corpus, shown in Table 6.4, indicate a general lower performance of the system on this corpus compared to those obtained on MPQA. A possible motivation to this lower performance may be found in the higher relative hardness of I-CAB Opinion with respect to MPQA, which can be hypothesized by observing the IAA values obtained on the two corpora, in which IAA is much higher on MPQA than on I-CAB Opinion (see Section 6.7.2).

In the comparison of the various feature set used on I-CAB Opinion, we have observed again an improvement in effectiveness by using SentiWordNet-based features with respect to the BASE feature set. The average improvement over the various tags is 3.56% for the SWN1 features= set and 3.39% for SWN2; values are lower than those measured on MPQA, most probably due the low coverage on the Italian language obtained by mapping SentiWordNet to this language using MultiWordNet. As for the MPQA experiments, it is the increase in recall which dominates the (eventual) loss in precision and determines the increase in overall effectiveness. The highest increase in performance is again observed on the expressive-subjectivity tag. It is worth to note that for the objective-speech-event tag the OE system scored a higher F_1 value (using the Token & Blank evaluation measure) than the one measured as IAA (see Table 6.5).

6.7.1 Statistical significance tests

In order to check whether the obtained results are statistically significant we have subjected them to thorough statistical significance testing.

A statistical significance test takes in input two classifications, generally produced by two independent classifiers, and outputs a probability value P that indicates the probability that the observed differences in the two classifications are due to chance. A low P value is thus an indication that the observed differences are significant (i.e. *not* due to chance) and that the classifiers that have produced them are substantially different. Two common threshold values for P used in literature are $P \leq 0.05$ and $P \leq 0.01$, identifying the recognition of a statistically significant difference in the compared experiments, with increasing confidence. When $P > 0.05$ the difference is considered to be *not* statistically significant.

We have applied to the results of our experiments the *s-test* and the *p-test*, two significance tests designed for text classification systems (see [100, Section 4]). The *s-test* is a sign test [85, Chapter 17] which compares two classifiers $\hat{\Phi}_1$ and $\hat{\Phi}_2$ by analyzing their binary decisions on each document/category pair. The *p-test*, on π and ρ, is a t-test which

Model	Annotation match									Token & Blank		
Predicate	Overlap			Head			Exact					
	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1
	AGENT											
BASE	.725	.526	.609	.652	.471	.547	.598	.432	.502	.634	.449	.526
GI	.715 (-1.33%)	.534 (1.61%)	.611 (0.35%)	.638 (-2.08%)	.476 (0.96%)	.545 (-0.34%)	.586 (-2.13%)	.436 (0.90%)	.500 (-0.39%)	.622 (-1.91%)	.457 (1.75%)	.527 (0.20%)
HM	.712 (-1.74%)	.538 (2.34%)	.613 (0.58%)	.638 (-2.15%)	.473 (0.41%)	.543 (-0.68%)	.586 (-2.13%)	.436 (0.90%)	.500 (-0.39%)	.622 (-1.95%)	.456 (1.59%)	.526 (0.09%)
SWN1	.711 (-1.82%)	.548 (4.21%)	.619 (1.59%)	.630 (-3.31%)	.484 (2.75%)	.548 (0.12%)	.578 (-3.45%)	.443 (2.58%)	.502 (-0.04%)	.624 (-1.65%)	.475 (5.90%)	.540 (2.63%)
SWN2	.713 (-1.63%)	.548 (4.29%)	.620 (1.71%)	.632 (-3.02%)	.484 (2.80%)	.548 (0.27%)	.578 (-3.35%)	.443 (2.42%)	.502 (-0.08%)	.621 (-2.05%)	.474 (5.52%)	.538 (2.24%)
ALLSUBJ	.701 (-3.21%)	.555 (5.52%)	.619 (1.66%)	.630 (-3.37%)	.487 (3.48%)	.550 (0.49%)	.565 (-5.67%)	.445 (3.03%)	.498 (-0.81%)	.623 (-1.79%)	.479 (6.72%)	.542 (3.02%)
Choi et al. [18]	.812	.606	.694	.793	.595	.680	.727	.541	.620	-	-	-
Choi et al. [17]	.806	.757	.781	-	-	-	-	-	-	-	-	-
	DIRECT-SUBJECTIVE											
BASE	.668	.424	.519	.555	.349	.428	.485	.305	.375	.613	.321	.422
GI	.664 (-0.59%)	.447 (5.45%)	.534 (3.02%)	.547 (-1.37%)	.365 (4.64%)	.438 (2.23%)	.476 (-2.04%)	.317 (3.95%)	.381 (1.55%)	.608 (-0.77%)	.341 (6.15%)	.437 (3.66%)
HM	.664 (-0.59%)	.447 (5.45%)	.534 (3.02%)	.540 (-2.65%)	.365 (4.64%)	.436 (1.70%)	.490 (0.93%)	.310 (1.51%)	.380 (1.29%)	.583 (-4.92%)	.330 (2.70%)	.421 (-0.06%)
SWN1	.660 (-1.19%)	.465 (9.76%)	.546 (5.23%)	.541 (-2.54%)	.377 (8.17%)	.444 (3.77%)	.472 (-2.88%)	.329 (7.81%)	.388 (3.42%)	.600 (-2.07%)	.358 (11.31%)	.448 (6.32%)
SWN2	.660 (-1.30%)	.464 (9.39%)	.545 (4.98%)	.539 (-2.79%)	.376 (7.71%)	.443 (3.40%)	.469 (-3.48%)	.327 (6.95%)	.385 (2.67%)	.599 (-2.36%)	.355 (10.57%)	.446 (5.75%)
ALLSUBJ	.654 (-2.20%)	.489 (15.31%)	.559 (7.82%)	.527 (-4.94%)	.391 (11.97%)	.449 (4.78%)	.460 (-5.25%)	.338 (10.60%)	.390 (3.89%)	.586 (-4.38%)	.378 (17.58%)	.460 (8.97%)
Breck et al. [11]	.722	.692	.707	-	-	-	.577	.427	.490	-	-	-
CRFs base [11]	.709	.461	.566	-	-	-	.451	.306	.364	-	-	-

Table 6.2: Results of automatic annotation of EPSs on the MPQA corpus, for AGENT and DIRECT-SUBJECTIVE tags of WWC markup language.

Model	Annotation match									Token & Blank		
Predicate	Overlap			Head			Exact					
	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1
	EXPRESSIVE-SUBJECTIVITY											
BASE	.668	.368	.474	.445	.230	.304	.234	.121	.159	.503	.293	.370
GI	.656 (-1.83%)	.384 (4.46%)	.484 (2.14%)	.422 (-5.23%)	.242 (4.82%)	.307 (1.17%)	.229 (-2.14%)	.129 (6.77%)	.165 (3.56%)	.499 (-0.72%)	.315 (7.46%)	.386 (4.29%)
HM	.658 (-1.45%)	.374 (1.74%)	.477 (0.58%)	.430 (-3.45%)	.238 (3.29%)	.306 (0.89%)	.229 (-2.14%)	.124 (2.63%)	.161 (0.96%)	.499 (-0.72%)	.315 (7.46%)	.386 (4.29%)
SWN1	.651 (-2.56%)	.414 (12.55%)	.506 (6.68%)	.433 (-2.76%)	.260 (12.65%)	.325 (6.88%)	.224 (-4.29%)	.134 (10.93%)	.168 (5.23%)	.500 (-0.58%)	.326 (11.38%)	.395 (6.65%)
SWN2	.652 (-2.34%)	.414 (12.61%)	.506 (6.81%)	.431 (-3.33%)	.258 (12.06%)	.323 (6.29%)	.225 (-3.64%)	.135 (11.85%)	.169 (6.04%)	.503 (0.02%)	.327 (11.52%)	.396 (6.99%)
ALLSUBJ	.637 (-4.66%)	.433 (17.65%)	.515 (8.63%)	.430 (-3.45%)	.263 (13.93%)	.326 (7.34%)	.223 (-4.61%)	.139 (15.05%)	.171 (7.50%)	.497 (-1.12%)	.335 (14.28%)	.400 (8.08%)
Breck et al. [11]	.638	.635	.634	-	-	-	.262	.190	.195	-	-	-
	OBJECTIVE-SPEECH-EVENT											
BASE	.556	.432	.486	.528	.410	.461	.503	.391	.440	.546	.372	.443
GI	.552 (-0.76%)	.438 (1.32%)	.488 (0.40%)	.520 (-1.45%)	.418 (1.96%)	.463 (0.44%)	.497 (-1.16%)	.391 (0.06%)	.438 (-0.48%)	.540 (-1.13%)	.380 (2.11%)	.446 (0.77%)
HM	.554 (-0.40%)	.435 (0.62%)	.487 (0.17%)	.525 (-0.50%)	.412 (0.49%)	.462 (0.05%)	.500 (-0.56%)	.395 (1.08%)	.441 (0.35%)	.540 (-1.13%)	.382 (2.65%)	.447 (1.08%)
SWN1	.550 (-1.06%)	.448 (3.63%)	.494 (1.53%)	.517 (-1.93%)	.421 (2.57%)	.464 (0.55%)	.491 (-2.32%)	.399 (2.18%)	.440 (0.16%)	.536 (-1.88%)	.390 (4.84%)	.452 (2.01%)
SWN2	.551 (-0.98%)	.448 (3.67%)	.494 (1.58%)	.519 (-1.58%)	.422 (3.04%)	.466 (0.97%)	.493 (-1.96%)	.401 (2.66%)	.442 (0.59%)	.537 (-1.67%)	.391 (5.03%)	.452 (2.21%)
ALLSUBJ	.546 (-1.89%)	.458 (5.98%)	.498 (2.39%)	.513 (-2.84%)	.430 (4.90%)	.468 (1.37%)	.485 (-3.48%)	.407 (4.22%)	.443 (0.71%)	.530 (-2.90%)	.400 (7.59%)	.456 (3.08%)

Table 6.3: Results of automatic annotation of EPSs on the MPQA corpus, for EXPRESSIVE-SUBJECTIVITY and OBJECTIVE-SPEECH-EVENT tags of WWC markup language.

Model	Annotation match									Token & Blank		
Predicate	Overlap			Head			Exact					
	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1	π	ρ	F_1
	AGENT											
BASE	.476	.235	.314	.442	.216	.291	.377	.184	.248	.397	.203	.269
SWN1	.470 (-1.37%)	.240 (2.09%)	.317 (0.92%)	.441 (-0.41%)	.222 (2.83%)	.296 (1.74%)	.370 (-1.87%)	.187 (1.33%)	.248 (0.26%)	.401 (0.88%)	.205 (0.92%)	.271 (0.90%)
SWN2	.463 (-2.72%)	.248 (5.48%)	.323 (2.63%)	.447 (1.04%)	.228 (5.48%)	.302 (3.98%)	.379 (0.60%)	.177 (-3.86%)	.248 (0.26%)	.400 (0.75%)	.205 (0.92%)	.271 (0.86%)
	DIRECT-SUBJECTIVE											
BASE	.466	.171	.250	.424	.155	.227	.424	.155	.227	.415	.124	.191
SWN1	.456 (-2.17%)	.177 (3.64%)	.255 (2.01%)	.416 (-1.82%)	.161 (4.00%)	.233 (2.37%)	.416 (-1.82%)	.161 (4.00%)	.233 (2.37%)	.403 (-2.71%)	.130 (4.41%)	.196 (2.68%)
SWN2	.447 (-4.13%)	.185 (8.38%)	.262 (4.72%)	.409 (-3.48%)	.158 (1.65%)	.228 (0.22%)	.412 (-2.87%)	.165 (6.46%)	.236 (3.78%)	.409 (-1.36%)	.130 (4.41%)	.197 (3.02%)
	EXPRESSIVE-SUBJECTIVITY											
BASE	.495	.222	.306	.411	.172	.243	.333	.139	.196	.407	.152	.221
SWN1	.499 (0.91%)	.245 (10.33%)	.328 (7.23%)	.420 (2.05%)	.194 (12.77%)	.266 (9.37%)	.362 (8.95%)	.168 (20.39%)	.229 (16.77%)	.409 (0.52%)	.170 (11.99%)	.240 (8.63%)
SWN2	.508 (2.69%)	.237 (6.92%)	.323 (5.57%)	.422 (2.61%)	.202 (17.39%)	.274 (12.60%)	.366 (9.94%)	.161 (15.58%)	.224 (13.86%)	.402 (-1.12%)	.169 (11.38%)	.238 (7.68%)
	OBJECTIVE-SPEECH-EVENT											
BASE	.592	.377	.460	.586	.372	.455	.579	.368	.450	.612	.383	.471
SWN1	.600 (1.33%)	.389 (3.33%)	.472 (2.55%)	.594 (1.37%)	.385 (3.37%)	.467 (2.58%)	.587 (1.41%)	.381 (3.41%)	.462 (2.62%)	.616 (0.66%)	.394 (2.88%)	.481 (2.01%)
SWN2	.600 (1.31%)	.392 (4.14%)	.474 (3.02%)	.596 (1.76%)	.378 (1.40%)	.462 (1.54%)	.595 (2.76%)	.389 (5.54%)	.470 (4.44%)	.616 (0.52%)	.394 (2.94%)	.481 (2.00%)

Table 6.4: Results of automatic annotation of EPSs on the I-CAB Opinion corpus.

compares two classifiers $\hat{\Phi}_1$ and $\hat{\Phi}_2$ by analyzing the microaveraged precision and recall values that the two systems have obtained. Both s-test and p-test are designed to compare two systems at the ("micro") level of individual classification decisions. Given the focus of our task on token classification, they are thus the ideal tests to evaluate the statistical significance of our experiments.

We have first compared all the experiments using the BASE features set with all the others, thus performing 21 statistical significance tests[20], obtaining a "highly statistically significant" judgment ($P \leq 0.01$) for 13 of them, a "statistically significant" judgment ($P \leq 0.05$) for 5, and a "not statistically significant" judgment for ($P > 0.05$) for the remaining 3 (all related to the GI or HM features sets). We can thus reaffirm what we hypothesized in the comments about the experiments, i.e., that the use of OM lexical resources has a significant impact on recall and the overall performance.

We have then compared the SWN1 and SWN2 experiments with the GI and HM experiments, obtaining, in the 12 tests, 6 "highly statistically significant" judgments, 3 "statistically significant" judgments, and 3 "not statistically significant" judgments. This is again a good support to state that SentiWordNet-based feature sets perform better than the other feature sets.

Last, we have compared the experiments based on the two versions of SentiWordNet. Unfortunately in this case 5 of the 6 tests answered with a "not statistically significant" judgment, leaving unanswered the question on which of the two versions is better when put in use. We like to point out that the results presented here do not conflict with those reported in Chapter 5, Sections 5.6 and 5.7. The results just state that the differences among SWN1 and SWN2 observed in our experiments may be just due to chance, and possibly not representative of a comparison between them.

6.7.2 Inter-Annotator Agreement

We report in this section the results of the IAA studies made on the two corpora, which can be useful for the interpretation of the overall performance obtained in our experiments.

[20] The five experiments using the subjectivity features for the MPQA corpus plus the two on the I-CAB Opinion corpus, all multiplied by the three significance tests performed: the s-test and the p-test on precision and recall.

Wiebe et al. [97] report an IAA study on the MPQA corpus. They have evaluated IAA by using the AGR measure. The AGR measure estimates the agreement between two independent annotators A and B by computing the arithmetic average between the precision, $\pi(A, B)$, and recall, $\rho(A, B)$, measures presented in Section 6.4.1 (Equations 6.1 and 6.2).

They report explicit IAA values only for two tags, out of the five defined in the WWC markup language. They have measured IAA by averaging the AGR measure calculated pairwise among three annotators, on a set of 13 documents. The measured agreement on the EXPRESSIVE-SUBJECTIVITY tag is 0.72, while on the DIRECT-SUBJECTIVE tag is 0.82.

The lower performance on the I-CAB Opinion corpus can be in part explained by looking at the IAA values reported in [35], which are relatively low with respect to those obtained on the MPQA corpus in [97]. The high agreement scored on the INSIDE tag has been one of the reasons that advise us to ignore it in our experiments, focusing on the hardest elements of the WWC markup language.

The IAA investigation on the I-CAB Opinion corpus has consisted in asking an intern (a third-year student in Computers and the Humanities) to independently annotate 127 (94 training and 33 test) articles (this accounts for 24% of the total 525 articles). The IAA is measured using various measures: the AGR measure presented in [95], already described, the F_1 measure computed using the overlap match predicate among annotations, Cohen's κ [19, 36], and F_1 computed using the Token & Blank model. Cohen's κ [19, 36] is a widely adopted IAA measure which is computed by the formula:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{6.4}$$

where $P(A)$ is the observed probability of agreement between the two annotators, and $P(E)$ is the probability of agreement by chance. $P(A)$ and $P(E)$ are typically estimated by using values in the contingency table (see Section 6.4) computed matching the annotations from the two annotators, and defining:

$$P(A) = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.5}$$

$$P(E) = \frac{(TP + FP)(TP + FN) + (TN + FP)(TN + FN)}{TP + TN + FP + FN} \tag{6.6}$$

	# annot.		Overlap		Token&Blank	
	A	B	AGR	F_1	κ	F_1
AGENT	1239	859	.539	.521	.439	.472
DIRECT-SUBJ.	263	246	.507	.507	.414	.422
EXPRESSIVE-SUBJ	924	467	.602	.537	.339	.357
OBJECTIVE-SPEECH-EV.	132	144	.501	.500	.462	.465
INSIDE	491	563	.767	.763	.718	.791

Table 6.5: IAA study on I-CAB Opinion. Number of annotations for the various tags (first 2 columns), and IAA results according to various IAA models (remaining columns).

The result of the IAA study are reported in Table 6.5; Annotator A is the third author of [35], while Annotator B is the intern.

6.7.3 The Overlap measure

In this section we report a simple study which shows the weakness of the annotation-based overlap measure we have described in Section 6.4.1.

Table 6.6 reports the performance results, measured using the overlap measure, obtained by an OE system that trivially expands all the annotations produced by our OE system, in the version using the BASE features set, by including in each annotation the N tokens following and preceding it. We have tested $N \in [0, 10]$.

For $N = 0$ we have the reference values also reported in Tables 6.2 and 6.3; for increasing values of N the precision, recall and F_1 values increase at an impressive rate. More relevantly, for values of $N \leq 3$ the $overlap_{avg}$ measure returns “reasonable” values, but the reported improvement could be more than 15%. Thus, the values returned by the overlap measure are not indicative of the real quality of the annotation.

We want to note also that the other measures described in Section 6.4 are not subject to this weakness: all the values returned by them drop to near-zero values just for $N = 2$. We can conclude that the results returned by the overlap measure can be considered reliable only if they are supported by other results from more robust measures.

N	Overlap match				$overlap_{avg}$
	π	ρ	F_1	(Δ%)	
AGENT					
0	*.725*	*.526*	*.609*		*1*
1	.736	.536	.620	(+1.78%)	1.02
2	.768	.571	.655	(+7.54%)	1.13
3	.789	.595	.678	(+11.33%)	1.19
4	.815	.630	.711	(+16.62%)	1.26
5	.845	.652	.736	(+20.78%)	1.3
6	.865	.687	.766	(+25.75%)	1.37
7	.877	.704	.781	(+28.20%)	1.42
8	.884	.721	.794	(+30.35%)	1.47
9	.898	.745	.814	(+33.60%)	1.52
DIRECT-SUBJECTIVE					
0	*.668*	*.424*	*.519*		*1*
1	.694	.443	.541	(+4.25%)	1.01
2	.734	.472	.575	(+10.77%)	1.05
3	.759	.491	.597	(+15.02%)	1.1
4	.795	.530	.636	(+22.58%)	1.17
5	.810	.559	.661	(+27.49%)	1.22
6	.818	.575	.675	(+30.13%)	1.26
7	.828	.599	.695	(+33.97%)	1.31
8	.845	.612	.710	(+36.84%)	1.39
9	.866	.628	.728	(+40.30%)	1.44
EXPRESSIVE-SUBJECTIVITY					
0	*.668*	*.368*	*.474*		*1.05*
1	.704	.402	.512	(+7.91%)	1.14
2	.729	.421	.533	(+12.48%)	1.23
3	.742	.442	.554	(+16.90%)	1.31
4	.761	.469	.581	(+22.46%)	1.38
5	.771	.489	.598	(+26.21%)	1.45
6	.788	.512	.621	(+30.89%)	1.54
7	.797	.530	.636	(+34.23%)	1.62
8	.801	.546	.650	(+37.00%)	1.67
9	.811	.559	.662	(+39.51%)	1.72
OBJECTIVE-SPEECH-EVENT					
0	*.556*	*.432*	*.486*		*1*
1	.575	.445	.502	(+3.13%)	1.03
2	.598	.473	.529	(+8.64%)	1.12
3	.610	.498	.548	(+12.71%)	1.19
4	.627	.530	.574	(+18.01%)	1.29
5	.641	.552	.593	(+21.85%)	1.34
6	.656	.567	.608	(+25.03%)	1.39
7	.671	.589	.627	(+28.95%)	1.45
8	.682	.603	.640	(+31.60%)	1.5
9	.692	.617	.653	(+34.13%)	1.56

Table 6.6: Results obtained by a system which expands its annotations by N tokens. Evaluation based on the BASE features set experiment on the MPQA corpus.

6.8 Conclusions

We have presented a comparative study on the use of OM lexical resources in an OE system. We have built a (relatively) simple OE system by using an SVM-based IE system and various sets of opinion-specific features. We have applied our OE system to the task of automatically annotating EPSs in documents. EPSs have been annotated in text using a markup language that identifies the agent holding the opinion/emotion, the opinion/emotion expression, and the agent that is the target of the opinion/emotion. We have run experiments on two annotated corpora on two languages, English and Italian.

In our experiments we have used various evaluation measures, divided into two classes: a class of annotation-based measures, commonly used in the literature, and a class of token-based measures, that we have originally proposed here. We deem that annotation-based measures have some relevant limitations (e.g. see Table 6.6), while token-based measures allow to perform a more rigorous evaluation of experiments, capturing all the aspects that determine the effectiveness of an IE system.

On the MPQA corpus, which is in English, we have performed a comparison of the OM lexical resources presented in the previous chapters, i.e., the lexicon from Hatzivassiloglou and McKeown [44] (HM), the one extracted from the General Inquirer [92] (GI), and the two versions of SentiWordNet presented in Chapters 4 and 5. Results show that the use of each OM lexical resource in the OE task produces a significant improvement with respect to a baseline system which does not use any OM lexical resource.

In the comparison of the effects of the various resources on the task we have measured a significantly better performance from SentiWordNet with respect to the HM and GI lexicons. Between the two versions of SentiWordNet there is no clear winner.

We have observed another interesting result by using all the resources in a single experiment, obtaining a significant improvement over each single resource. This result suggests that none of the various resources tested does completely “include” the others, indicating that there is still room for their improvement.

On the I-CAB Opinion corpus, which is in Italian, we have tested the possibility of converting to another language the SentiWordNet scores, by using MultiWordNet [78] to map English synsets scores to Italian. We have obtained a significant improvement in performance, although smaller than the one observed on English, most probably due to: (a) the relatively small coverage of MultiWordNet of the Italian language, (b) the probably higher hardness of the OE task on Italian, as pointed out by the low IAA values on the I-CAB Opinion corpus.

The results we have obtained do not represent the current state of the art in OE. This fact has various motivations, the principal three being (a) the learning algorithm adopted, (b) the text processing tools used to build representations of tokens, and (c) the independent analysis of each tag type. In this first approach to the OE problem we have preferred to build a rather simple system and to concentrate on the comparison of OM lexical resources. Future developments of the system should focus on the points (a), (b) and (c) mentioned above, through:

- Use of the emerging class of *Conditional Random Fields* algorithms as the learning device for OE. This type of algorithms have shown to give good results in IE and OE tasks [18, 59].

- Use of robust NLP tools to enrich the token representation, e.g., the CASS partial parser [1], as used in [11, 55].

- Inclusion of more lexical resources. For example, adding the "subjectivity clues" patterns produced by Riloff and Wiebe [81], by using bootstrapping methods on unlabeled data. A match of one of such patterns with a part of text can be used as a clue that a subjective expression is present in the matched part.

- Performing a joint extraction of all the various components of opinion expressions, so that the information related to a component could support the identification of the others.

Chapter 7

Conclusion

7.1 Summary of results

In this thesis we have investigated several tasks related to the automatic generation of lexical resources for OM.

We have started from the (relatively) simple problem of determining term orientation, then we have moved to more complex problems, such as recognizing the subjectivity of terms, and then shifting down to the analysis of the distinct senses of terms, investigating various approaches to the problem. We finally have evaluated and compared the impact of various lexical resources, those already available in literature and those we have created, in a practical Opinion Extraction application.

A common trait of all the various methods we have proposed in this thesis is that they all exploit the information contained in the *glosses* associated to the various term senses in a dictionary, to gather the relevant information about the various dimensions of ORPs they work on.

We have proved that glosses, which are a resource by definition devoted to human comprehension, can be also productively used in automatic learning processes. One of the key advantages deriving from the use of glosses is that they are already available for almost any language, thus allowing a rapid application of any gloss-based method to many languages with reduced work.

In Chapter 1 we have defined a semi-supervised term classification method that takes in input a number of small seed sets, each one representing one of the ORPs to be automatically determined. Thus, one of the advantages of our method is that is requires a minimal human labeling effort to produce the input data. On the task of determining the orientation of terms our method has produced state-of-the-art results, by using as input just the two terms `good` and `bad`.

The method has shown to be easily adaptable to other tasks, such as the ones of classifying terms by their subjectivity (Chapter 2) or their attitude type (Chapter 3), continuing to produce good results in experiments with respect to the relative hardness of the tasks.

With SentiWordNet 1.0 (and 1.1) (Chapter 4), we have shown a further adaptation of our method to classify *term senses*, represented by WordNet synsets. SentiWordNet is a very unique resource, for many reasons:

- Its term-sense level of analysis is the most detailed one ever used in producing an OM lexical resource.
- Its graded-score model allows capturing all the subtle nuances of ORPs in term senses.
- It has a practically complete coverage of the English language, i.e. the same coverage of WordNet (version 2.0); this large coverage largely compensates the errors SentiWordNet contains, given its automatic generation method, as experiments on OE have also shown.
- The alignment of SentiWordNet with a widely used resource like WordNet makes it easily integrable with the many applications based on WordNet.
- The alignment of SentiWordNet with WordNet makes it also easily portable to other languages, leveraging on the various WordNet-aligned resources on other languages[1], as we have effectively shown in Chapter 6.

In Chapter 5 we have shown an effective application of random-walk models to the determination of ORPs of term senses. Again, the key input source of information have been the glosses of term senses, but analyzed from a different, “orthogonal” perspective than the one adopted in

[1] The site of the Global WordNet Association (`http://www.globalwordnet.org/`) currently lists about 50 projects developing WordNet resources on many languages.

the term classification method. In the term classification method glosses are used to *describe* term senses (or just terms) in a vector space, and such description is used to identify similar term senses (or terms). In the random-walk method there is no such notion of term sense description, but instead glosses are used to establish direct *relations* among term senses.

We have investigated three models: direct, inverse, and bidirectional. We have proved how all these models can be efficiently implemented as variants of the PageRank algorithm.

Our use of random-walk models, especially the inverted one, has proven capable of capturing the ORPs of term senses. Moreover, it has shown the ability to improve the already good-quality data (i.e., SentiWordNet 1.0) fed as input to describe such ORPs. We have experimentally shown this in two ways:

- by comparing the positivity and negativity rankings given as input with those obtained as output by the various random-walk models;
- by producing a new version of SentiWordNet (2.0) which improves over the previous one.

The generation of SentiWordNet 2.0 has also the merit of showing how the rankings produced by our random-walk algorithm can be successfully converted into a usable lexical resource.

Finally, in Chapter 6 we have compared the two main lexical resources we have generated, SentiWordNet 1.0 and 2.0, with other two well-known OM lexical resources, in an opinion extraction task.

Results have shown that SentiWordNet produces significantly better results than the other two resources, in both versions. This confirms the intuition that the high coverage of the language provided by SentiWordNet largely compensates its errors. Moreover, all the resources have produced significant improvement over the baseline system not using any OM lexical resource, indicating the relevance of such resources for obtaining high-quality results.

We have also shown an effective cross-language use of SentiWordNet, by running experiments on opinion extraction on the Italian language, using an Italian "translation" of SentiWordNet scores based on a WordNet-aligned Italian lexical resource, MultiWordNet [78], obtaining again significant improvements.

7.2 Future research directions

From the experience collected during the various challenges we have faced in this thesis we may draw some final thoughts on the future relevant directions of the research in this field.

With respect to the research on the generation of lexical resources for OM, which is the main topic of this thesis, we are deeply convinced that the future researches on the topic have to be focused on the analysis of ORPs of *term senses*.

In all the thesis we have dealt with the recognition of ORPs of terms, or term senses, in an "out of context" fashion. At the term level of analysis this could lead to some unacceptable simplifications, as already mentioned in Chapter 4. For example the term `pretty` is generally considered to be a positive term, but in the expression "`A pretty mess`" its valence is fully negative. Fortunately this phenomenon happens much more rarely at the term sense level where, for example, the ironic (and negative) sense of `pretty` is distinguished[2] from the other (positive) ones.

The term sense level of analysis allows also to capture opinion-related differences deriving from the special use of a term in a specific domain. For example, the name `dog` is usually used to refer to the animal and has no connotation, but when it is used to refer to a person it has instead a negative valence, indicating someone regarded as contemptible, and this distinction is typically reported in any dictionary.

Our experience with the complex attitude taxonomy, in Chapter 3, has shown how many other interesting dimensions contribute to the definition of an opinion expression. Although the generation of lexical resources on such dimension has demonstrated to be a rather hard task, we deem that the availability of a SentiWordNet-like resource on such dimensions could give a relevant boost to OM application performance.

With respect to the two main methods we have proposed in this thesis we envisage their direct application to other languages, using as sources of glosses and relations among terms one, or more, of the many electronic version of dictionaries currently available online[3]. It would be probably easier to port to other languages the term classification method

[2] We here specifically refer to WordNet but almost any dictionary reports such distinction.

[3] For example, for the Italian language `http://www.demauroparavia.it`, for French `http://www.cnrtl.fr/`.

than porting the random-walk algorithm, given the need of the latter for sense-disambiguated glosses.

We also foresee two rather imminent updates for SentiWordNet that consist in generating a version based on the last WordNet version (3.0) and repeating our random-walk models experiments as soon as the manually sense-disambiguated glosses for that version will be released.

Finally, with respect to the idea of using random-walk models to determine ORPs of term senses, we would like to investigate further on the problem, studying the application of our algorithms to other dimensions of term sense semantics, e.g., recognition of membership in domains or measuring sense similarity. We are also interested into adapting our random-walk algorithms to perform word sense disambiguation.

Bibliography

[1] Steven Abney. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344, 1996.

[2] Alina Andreevskaia and Sabine Bergler. Mining WordNet For a Fuzzy Sentiment: Sentiment Tag Extraction From WordNet Glosses. In *Proceedings EACL-06, the 11rd Conference of the European Chapter of the Association for Computational Linguistics*, pages 209–216, Trento, IT, 2006.

[3] Alina Andreevskaia and Sabine Bergler. Sentiment Tagging of Adjectives at the Meaning Level. In *Canadian Conference on AI*, pages 336–346, 2006.

[4] Shlomo Argamon, Bloom Kenneth, Andrea Esuli, and Fabrizio Sebastiani. Automatically Determining Attitude Type and Force for Sentiment Analysis. In *Proceedings of LTC-07, the 3rd Language & Technology Conference*, Poznan, PL, October 2007. To appear.

[5] Giuseppe Attardi and Maria Simi. Blog Mining Through Opinionated Words. In *Proceedings of TREC 2006, the Fifteenth Text Retrieval Conference*, Gaithersburg , US, 2006. NIST.

[6] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 86–90, San Francisco, California, 1998. Morgan Kaufmann Publishers.

[7] Marco Baroni and Stefano Vegnaduzzo. Identifying Subjective

Adjectives through Web-based Mutual Information. In *Proceedings of KONVENS-04*, pages 17–24, Vienna, AU, 2004.

[8] Luisa Bentivogli and Christian Girardi Emanuele Pianta. The MEANING Italian corpus. In *Proceedings of the Conference on Corpus Linguistics (CL'03)*, pages 103–112, Lancaster, UK, 2003.

[9] Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Automatic Extraction of Opinion Propositions and their Holders. In James G. Shanahan, Janyce Wiebe, and Yan Qu, editors, *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Stanford, US, 2004.

[10] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 5(1):92–128, 2005.

[11] Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*, Hyderabad, India, January 2007.

[12] Eric Brill. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistic*, 21(4):543–565, 1995. ISSN 0891-2017.

[13] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[14] Amedeo Cappelli and Bernardo Magnini, editors. *Proceedings of the 1st Workshop on the Evaluation of NLP Tools for Italian (EVALITA'07)*, Roma, IT, 2007.

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

[16] N. Chinchor. Overview of MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, Fairfax, Virginia, 1998.

[17] Yejin Choi, Eric Breck, and Claire Cardie. Joint Extraction of Entities and Relations for Opinion Recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, 2006.

[18] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *Proceedings of HLT-EMNLP-05, the Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Vancouver, CA, 2005.

[19] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[20] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* John Wiley & Sons, New York, US, 1991.

[21] Hamish Cunningham. GATE, a general architec ture for text engineering. *Computers and the Humani ties*, 36(2):223–254, 2002.

[22] Sanjiv R. Das and Mike Y. Chen. Yahoo! for Amazon: Sentiment Parsing from Small Talk on the Web. In *Proceedings of EFA 2001, European Finance Association Annual Conference*, Barcelona, ES, 2001.

[23] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW-03, 12th International Conference on the World Wide Web*, pages 519–528, Budapest, HU, 2003. ACM Press. ISBN 1-58113-680-3.

[24] Ann Devitt and Carl Vogel. The Topology of WordNet: Some Metrics. In *Proceedings of GWC-04, 2nd Global WordNet Conference*, pages 106–111, Brno, CZ, 2004.

[25] Xiaowen Ding and Bing Liu. The utility of linguistic rules in opinion mining. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in*

information retrieval, pages 811–812, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7.

[26] Andrea Esuli. The Sentiment Classification Bibliography. http://liinwww.ira.uka.de/bibliography/Misc/Sentiment.html. URL `http://liinwww.ira.uka.de/bibliography/Misc/Sentiment.html`.

[27] Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. MP-Boost: A Multiple-Pivot Boosting Algorithm and Its Application to Text Categorization. In *Proceeding of SPIRE-06, the 13th International Conference on String Processing and Information Retrieval*, volume 4209 of *Lecture Notes in Computer Science*, pages 1–12, Glasgow, UK, 2006. Springer. ISBN 3-540-45774-7.

[28] Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. Tree-Boost.MH: A Boosting Algorithm for Multi-label Hierarchical Text Categorization. In *Proceeding of SPIRE-06, the 13th International Conference on String Processing and Information Retrieval*, volume 4209 of *Lecture Notes in Computer Science*, pages 13–24, Glasgow, UK, 2006. Springer. ISBN 3-540-45774-7.

[29] Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. Bosting Multi-label Hierarchical Text Categorization. *Information Retrieval*, Forthcoming.

[30] Andrea Esuli and Fabrizio Sebastiani. Determining the semantic orientation of terms through gloss classification. In *Proceedings of CIKM-05, the ACM SIGIR Conference on Information and Knowledge Management*, pages 617–624, Bremen,DE, 2005. ACM Press.

[31] Andrea Esuli and Fabrizio Sebastiani. Determining Term Subjectivity and Term Orientation for Opinion Mining. In *Proceedings EACL-06, the 11rd Conference of the European Chapter of the Association for Computational Linguistics*, pages 193–200, Trento, IT, 2006.

[32] Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, Genova, IT, 2006.

[33] Andrea Esuli and Fabrizio Sebastiani. PageRanking WordNet Synsets: An Application to Opinion Mining. In *Proceedings of ACL-07, the 45th Annual Meeting of the Association of Computational Linguistics*, pages 424–431, Prague, CZ, June 2007. Association for Computational Linguistics.

[34] Andrea Esuli and Fabrizio Sebastiani. Random-Walk Models of Term Semantics: An Application to Opinion-Related Properties. In *Proceedings of LTC-07, the 3rd Language & Technology Conference*, Poznan, PL, October 2007. To appear.

[35] Andrea Esuli, Fabrizio Sebastiani, and Ilaria Clara Urciuoli. Annotating Expressions of Opinion and Emotion in the Italian Content Annotation Bank. In *Proceedings of LREC-08, the 6th Conference on Language Resources and Evaluation*, Marrakech, MR, 2008.

[36] Barbara Di Eugenio and Michael Glass. The kappa statistic: A second look. *Computational Linguistics*, 30(1):95–101, 2004.

[37] David Kirk Evans, Lun-Wei Ku, Yohei Seki, Hsin-Hsi Chen, and Noriko Kando. Opinion Analysis Across Languages: An Overview of and Observations from the NTCIR6 Opinion Analysis Pilot Task. In *7th International Workshop on Fuzzy Logic and Applications*, volume 4578 of *Lecture Notes in Computer Science*, pages 456–463. Springer, 2007.

[38] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing and aggregating rankings with ties. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58, New York, NY, USA, 2004. ACM. ISBN 158113858X.

[39] Zhongchao Fei, Jian Liu, and Gengfeng Wu. Sentiment Classification Using Phrase Patterns. In *Proceedings of CIT-04, the 4th International Conference on Computer and Information Technology*, pages 1147–1152, Wuhan, CN, 2004.

[40] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.

[41] Dayne Freitag. Using grammatical inference to improve precision in information extraction. In *Workshop on Grammatical Inference, Automata Induction, and Language Acquisition (ICML'97)*, Nashville, TN, 1997.

[42] Gregory Grefenstette, Yan Qu, James G. Shanahan, and David A. Evans. Coupling Niche Browsers and Affect Analysis for an Opinion Mining Application. In *Proceeding of RIAO-04*, Avignon, FR, 2004.

[43] Sanda M. Harabagiu, George A. Miller, and Dan I. Moldovan. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In *Proceedings of the SIGLEX Workshop*, 1999.

[44] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181, Madrid, ES, 1997. Association for Computational Linguistics.

[45] Vasileios Hatzivassiloglou and Janyce M. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of COLING-00, 18th International Conference on Computational Linguistics*, pages 299–305, Saarbrücken, GE, 2000. Morgan Kaufmann.

[46] Taher H. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

[47] Thad Hughes and Daniel Ramage. Lexical Semantic Relatedness with Random Graph Walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 581–589, 2007.

[48] Nancy Ide. Making Senses: Bootstrapping Sense-tagged Lists of Semantically-Related Words. In *Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING'06)*, pages 13–27, Mexico City, MX, 2006.

[49] Neil Ireson, Fabio Ciravegna, Mary Elaine Califf, Dayne Freitag, Nicholas Kushmerick, and Alberto Lavelli. Evaluating machine learning for information extraction. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 345–352, New York, US, 2005. ACM. ISBN 1-59593-180-5.

[50] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[51] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the "Lecture Notes in Computer Science" series, number 1398.

[52] Jaap Kamps and Maarten Marx. Words with attitude. In *Proceedings of the 1st International Conference on Global WordNet*, pages 332–341, Mysore, IN, 2002.

[53] Jaap Kamps, Maarten Marx, R. ort. Mokken, and Maarten de Rijke. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation*, volume IV, pages 1115–1118, Lisbon, PT, 2004.

[54] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In *Proceedings COLING-04, the Conference on Computational Linguistics*, Geneva, CH, 2004.

[55] Soo-Min Kim and Eduard Hovy. Identifying Opinion Holders for Question Answering in Opinion Texts. In *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*, Pittsburgh, US, 2005.

[56] Soo-Min Kim and Eduard Hovy. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*, Sidney, AUS, 2006.

[57] Taku Kudo and Yuji Matsumoto. Use of support vector learning for chunk identification. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, pages 142–144, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[58] Taku Kudo and Yuji Matsumoto. Fast methods for kernel-based text analysis. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 24–31, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[59] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[60] A. Lavelli, M. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, and L. Romano. IE evaluation: Criticisms and recommendations. In *Proceedings of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining (ATEM 2004)*, San Jose, US, 2004.

[61] Michael E. Lesk. Automatic word sense disambiguation: How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC-86, 5th ACM International Conference on Systems Documentation*, pages 24–26, New York, US, 1986.

[62] David D. Lewis. Evaluating and optmizing autonomous text classification systems. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, US, 1995. ACM Press, New York, US.

[63] Bernardo Magnini and Gabriela Cavaglià. Integrating Subject Field Codes into WordNet. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'2000)*, pages 1413–1418, Athens, GR, 2000.

[64] Bernardo Magnini, Emanuele Pianta, Christian Girardi, Matteo Negri, Lorenza Romano, Manuela Speranza, Valentina Bartalesi-Lenzi, and Rachele Sprugnoli. I-CAB: The Italian content annotation bank. In *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, pages 963–968, Genova, IT, 2006.

[65] J. R. Martin and P. R. R. White. *The Language of Evaluation: Appraisal in English.* Palgrave, London, UK, 2005.

[66] Andrew K. McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*, pages 41–48, Madison, US, 1998.

[67] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. PageRank on semantic networks, with application to word sense disambiguation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1126, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[68] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the Web. In *Proceedings of KDD-02, 8th ACM International Conference on Knowledge Discovery and Data Mining*, pages 341–349, Edmonton, CA, 2002. ACM Press. ISBN 1-58113-567-X.

[69] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms.* Cambridge University Press, Cambridge, UK, 1995.

[70] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP-04, 9th Conference on Empirical Methods in Natural Language Processing*, Barcelon, ES, 2004.

[71] Tetsuya Nasukawa and Jeonghee Yi. Sentiment Analysis: Capturing Favorability using Natural Language Processing. In *Proceedings of the K-CAP-03, 2nd International Conference on Knowledge Capture*, pages 70–77, New York, US, 2003. ACM Press. ISBN 1-58113-583-1.

[72] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[73] Charles E. Osgood, George J. Suci, and Percy H. Tannenbaum. *The Measurement of Meaning.* University of Illinois Press, 1967.

[74] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 Blog Track. In *Proceeddings of the 15th Text REtrieval Conference (TREC 2006)*, 2006.

[75] Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on Minimum Cuts. In *Proceedings of ACL-04, 42nd Meeting of the Association for Computational Linguistics*, pages 271–278, Barcelona, ES, 2004. Association for Computational Linguistics.

[76] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL-05, 43nd Meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, US, 2005. Association for Computational Linguistics.

[77] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of EMNLP-02, the Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, US, 2002. Association for Computational Linguistics.

[78] Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. MultiWordNet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, Mysore, IN, 2002.

[79] Ellen Riloff. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, Vol. 2*, pages 1044–1049, Portland, US, 1996.

[80] Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. Feature Subsumption for Opinion Analysis. In *Proceedings of EMNLP-06, the Conference on Empirical Methods in Natural Language Processing*, pages 440–448, Sydney, AUS, July 2006. Association for Computational Linguistics.

[81] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CONLL-03, 7th Conference on Natural Language Learning*, pages 25–32, Edmonton, CA, 2003.

[82] Carl Sable and Ken Church. Using Bins to Empirically Estimate Term Weights for Text Categorization. In Lillian Lee and Donna Harman, editors, *Proceedings of EMNLP-01, 6th Conference on Empirical Methods in Natural Language Processing*, pages 58–66, Pittsburgh, US, 2001. Association for Computational Linguistics, Morristown, US.

[83] Carl Sable, Kathleen McKeown, and Kenneth W. Church. NLP Found Helpful (at least for one Text Categorization Task). In *Proceedings of EMNLP-02, Conference on Empirical Methods in Natural Language Processing*, pages 172–179, Philadelphia, US, 2002. Association for Computational Linguistics.

[84] Tjong Kim Sang, Erik F., and Sabine Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Lisbon, Portugal, 2000.

[85] Murray R. Spiegel and Larry J. Stephens. *Statistics.* McGraw-Hill, New York, US, third edition, 1999.

[86] 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.

[87] Pero Subasic and Alison Huettner. Affect analysis of text using fuzzy semantic typing. *IEEE Transaction on Fuzzy Systems*, 9(4): 483–496, 2001.

[88] Maite Taboada and Jack Grieve. Analyzing Appraisal Automatically. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 158–161, Stanford, US, 2004.

[89] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting Emotional Polarity of Words using Spin Model. In *Proceedings of ACL-05, 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, US, 2005. Association for Computational Linguistics.

[90] Kagan Tumer and Joydeep Ghosh. Error Correlation and Error Reduction in Ensemble Classifiers. *Connection Science*, 8(3-4): 385–403, 1996.

[91] Peter Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of ACL-02, 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, US, 2002. Association for Computational Linguistics.

[92] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346, 2003. ISSN 1046-8188.

[93] Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. Developing Affective Lexical Resources. *PsychNology Journal*, 2(1): 61–83, 2004.

[94] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using Appraisal Taxonomies for Sentiment Analysis. In *Proceedings of MCLC-05, the 2nd Midwest Computational Linguistic Colloquium*, Columbus, US, 2005.

[95] Janyce Wiebe, E. Breck, Christopher Buckley, Claire Cardie, P. Davis, B. Fraser, Diane Litman, D. Pierce, Ellen Riloff, Theresa Wilson, D. Day, and Mark Maybury. Recognizing and Organizing Opinions Expressed in the World Press. In *Proceedings of the 2003 AAAI Spring Symposium on New Directions in Question Answering*, 2003.

[96] Janyce Wiebe and Rada Mihalcea. Word sense and subjectivity. In *Proceedings of COLING/ACL-06, the 21st Conference on Computational Linguistics / Association for Computational Linguistics*, pages 1065–1072, Sydney, AUS, 2006. Association for Computational Linguistics.

[97] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 1(2):0–0, 2005.

[98] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of AAAI-04, 21st Conference of the American Association for Artificial Intelligence*, pages 761–769, San Jose, US, 2004. AAAI Press / The MIT Press.

[99] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Recognizing Strong And Weak Opinion Clauses. *Computational Intelligence*, 22(2):73–99, May 2006.

[100] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 42–49, Berkeley, US, 1999.

[101] Hong Yu and Vasileios Hatzivassiloglou. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In Michael Collins and Mark Steedman, editors, *Proceedings of EMNLP-03, 8th Conference on Empirical Methods in Natural Language Processing*, pages 129–136, Sapporo, JP, 2003.

[102] S. M. Zeno, S. H. Ivens, R. T. Millard, and R. Duvvuri. *The Educator's Word Frequency Guide.* Touchstone Applied Science Associates, Inc., New York, US, 1995. ISBN 1564970213.

List of Figures

List of Tables