Determining the Semantic Orientation of Terms through Gloss Classification

Andrea Esuli
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
Via G Moruzzi, 1 – 56124 Pisa, Italy
andrea.esuli@isti.cnr.it

Fabrizio Sebastiani
Dipartimento di Matematica Pura e Applicata
Università di Padova
Via GB Belzoni, 7 – 35131 Padova, Italy
fabrizio.sebastiani@unipd.it

ABSTRACT

Sentiment classification is a recent subdiscipline of text classification 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 *orientation* of "subjective" terms contained in text, i.e. the determination of whether a term that carries opinionated content has a positive or a negative connotation. In this paper we present a new method for determining the orientation of subjective terms. The 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 method we present outperforms all known methods when tested on the recognized standard benchmarks for this task.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering; Search process*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

General Terms

Algorithms, Experimentation

Keywords

Opinion Mining, Text Classification, Semantic Orientation, Sentiment Classification, Polarity Detection

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

1. INTRODUCTION

Text classification (TC) is the task of automatically attributing a document d_i to zero, one or several among a predefined set of categories $C = \{c_1, \ldots, c_n\}$ based on the analysis of the contents of d_i. Throughout the history of TC, *topic-relatedness* (aka *thematic affinity*, or *aboutness*) has been the main dimension in terms of which TC has been studied, with categories representing topics and classification coinciding with the assignment to c_j of those documents that were deemed to be *about* topic c_j.

With the improvement of TC technology, and with the ensuing increase in the effectiveness and efficiency of text classifiers, new (and less obvious) dimensions "orthogonal" to topic-relatedness have started to be investigated. Among these, of particular relevance are *genre classification*, as in deciding whether a given product description is a Review or an Advertisement; *author classification* (aka *authorship attribution*), as in deciding who, among a predefined set of candidate authors, wrote a given text of unknown or disputed paternity; and *sentiment classification*, as in deciding whether a given text expresses a positive or a negative opinion about its subject matter. It is this latter task that this paper focuses on.

In the literature, sentiment classification [4, 14] also goes under different names, among which *opinion mining* [2, 5, 11], *sentiment analysis* [12, 13], *sentiment extraction* [1], or *affective rating* [3]. It has been an emerging area of research in the last years, largely driven by applicative interest in domains such as mining online corpora for opinions, or customer relationship management. Sentiment classification can be divided into several specific subtasks:

1. *determining subjectivity*, as in deciding whether a given text has a factual nature (i.e. describes a given situation or event, without expressing a positive or a negative opinion on it) or expresses an opinion on its subject matter. This amounts to a binary classification task under categories Objective and Subjective [13, 20];

2. *determining orientation* (or *polarity*), as in deciding whether a given Subjective text expresses a Positive or a Negative opinion on its subject matter [13, 17];

3. *determining the strength of orientation*, as in deciding e.g. whether the Positive opinion expressed by a text on its subject matter is Weakly Positive, Mildly Positive, or Strongly Positive [19].

Functional to all these tasks[1] is the determination of the orientation of individual terms present in the text, such as determining that (using Turney and Littman's [18] examples) `honest` and `intrepid` have a positive connotation while `disturbing` and `superfluous` have a negative connotation, since it is by considering the combined contribution of these terms that one may hope to solve Tasks 1, 2 and 3. The conceptually simplest approach to this latter problem is probably Turney's [17], who has obtained interesting results on Task 2 by considering the algebraic sum of the orientations of terms as representative of the orientation of the document they belong to; but more sophisticated approaches are also possible [7, 15, 19].

We propose 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 Positive or Negative. 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 `honest` and `intrepid` will both contain appreciative expressions, while the glosses of `disturbing` and `superfluous` will both contain derogative expressions. The method is semi-supervised, 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 algorithm on the three benchmarks previously used in this literature, and first proposed in [6, 9, 18], respectively. Our method is found to outperform the previously known best-performing method [18] 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 Outline of the paper

In Section 2 we review in some detail the related literature on determining the orientation of terms. The methods and results presented in this section are analysed and taken as reference in Section 3, which describes our own approach to determining the orientation of terms, and in Sections 4 and 5, which report on the experiments we have run and on the results we have obtained. Section 6 concludes.

[1]Task 1 may be seen as being subsumed by Task 2 in case this latter also includes a Neutral category. Similarly, Task 2 may be seen as being subsumed by Task 3 in case this latter contains an ordered sequence of categories ranging from Strongly Negative to Neutral to Strongly Positive.

2. RELATED WORK

2.1 Hatzivassiloglou and McKeown [6]

The work of Hatzivassiloglou and McKeown [6] 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 analysing pairs of adjectives (conjoined by `and`, `or`, `but`, `either-or`, or `neither-nor`) extracted from a large unlabelled 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 [6]:

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. 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.
3. A clustering algorithm uses the graph produced in Step 2 to partition the adjectives into two clusters. 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.

For their experiments, the authors used a term set consisting of 657/679 adjectives labelled as being Positive/Negative (hereafter, *the HM term set*). The document collection from which they extracted the conjunctions of adjectives is the unlabelled 1987 Wall Street Journal document set[2]. In the experiments reported in [6], the above algorithm determines the orientation of adjectives with an accuracy of 78.08% on the full HM term set.

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

2.2 Turney and Littman [18]

Turney and Littman [18] have approached the problem of determining the orientation of terms by bootstrapping from a pair of two minimal sets of "seed" terms (hereafter, we will call such a pair a *seed set*):

- 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 categories Positive and Negative. Their method is based on computing the *pointwise mutual information* (PMI)

$$PMI(t, t_i) = \log \frac{\Pr(t, t_i)}{\Pr(t)\Pr(t_i)} \quad (1)$$

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 $O(t)$ (where positive value means positive orientation, and higher absolute value means stronger orientation) is given by

$$O(t) = \sum_{t_i \in S_p} PMI(t, t_i) - \sum_{t_i \in S_n} PMI(t, t_i) \quad (2)$$

The authors have tested their method on the HM term set from [6] and also on the categories Positive and Negative defined in the General Inquirer lexicon [16]. 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 polarity. 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. 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 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"; this amounted to 61,000 documents, for a total of about 10 million word occurrences.

The results of [18] 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 [6]. 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 [6]. Using AND instead of NEAR on AV-Eng brought about a 19% decrease in accuracy with respect 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).

2.3 Kamps et al. [9]

Kamps et al. [9] focused on the use of lexical relations defined in WordNet (WN)[6]. They defined a graph on the adjectives contained in the intersection between the TL term set and WN, adding a link between two adjectives whenever WN indicates the presence of a synonymy relation between them. On this graph, the authors defined a 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.

$$SO(t) = \frac{d(t, \texttt{bad}) - d(t, \texttt{good})}{d(\texttt{good}, \texttt{bad})} \quad (3)$$

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 WN 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.

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

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

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

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

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 process is composed of the following steps:

1. A seed set (S_p, S_n), representative of the two categories Positive and Negative, is 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 S_p and S_n. This process can be iterated. The new terms, once added to the original ones, yield two new, richer sets S'_p and S'_n of terms; together they form the training set for the learning phase of Step 4.

3. For each term t_i in $S'_p \cup S'_n$ 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 $S'_p \cup S'_n$ and then applied to the terms in the test set.

Step 2 is based on the hypothesis that the lexical relations used in this expansion phase, in addition to defining a relation of meaning, 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[8]. This step is thus reminiscent of the use of the synonymy relation as made by Kamps et al. [9]. 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, the basic assumption is that terms with a similar orientation tend to have "similar" glosses: for instance, that the glosses of `honest` and `intrepid` will contain both appreciative expressions, while the glosses of `disturbing` and `superfluous` will contain both derogative expressions. 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[9].

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

[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.

[8]This intuition is basically the same as that of Kim and Hovy [10], whose paper was pointed out to us at the time of going to press.

[9]Experiments in which some unintended senses and their glosses are filtered out by means of part-of-speech analysis are described in Section 5.

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: (i) taking advantage of the richness and precision of human-defined linguistic characterizations as available in lexical resources such as WordNet; and (ii) enabling the classification of *any* term, provided there is a gloss for it in the lexical resource. 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 [6], which can only be applied to adjectives, and with that of [9], which can only be applied to terms directly or indirectly connected to the terms `good` or `bad` through the WordNet synonymy relation.

4. EXPERIMENTS

4.1 Test sets and seed sets

We have run our experiments on the HM, TL, and KA term sets, described in Sections 2.1, 2.2, and 2.3, respectively. As discussed in Section 3, the method requires bootstrapping from a seed set (S_p, S_n) representative of the categories Positive and Negative. In the experiments we have alternatively used the same seven positive and seven negative terms used in [18] (the *Tur* training set), as listed in Section 2, or the singleton sets {`good`} and {`bad`} (the *Kam* training set), as used in [9]. Note that *Kam* is a proper subset of *Tur*.

4.2 Expansion method for seed sets

We have used WordNet version 2.0 (WN)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 WN, 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`)[10], *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 used for expanding (S_p, S_n), is described in Figure 1. The input parameters are the initial seed set (S_p, S_n) 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 set (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)[11]. The role of Steps 3 and 4 is to avoid that the same term be added to both S_p and S_n; this is accomplished by applying the two rules of *Priority*

[10]Indirect antonymy is defined in WN 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 is mediated by the similarity of `parched` and `dry`. It should be remarked that $Ant^D \subseteq Ant^I$.

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

function *ExpandSimple*

Input :

(S_p, S_n) : seed set 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 :

(S'_p, S'_n) : expanded seed set

Body :

1. $S'_p \leftarrow S_p$; $S'_n \leftarrow S_n$;
2. foreach $term$ in S_p do
   $Temp \leftarrow$ set of all terms directly connected to $term$ in G_{rel};
   if S_{rel} then
   $S'_p \leftarrow S'_p \cup Temp$;
   else
   $S'_n \leftarrow S'_n \cup Temp$;
   foreach $term$ in S_n do
   $Temp \leftarrow$ set of all terms directly connected to $term$ in G_{rel};
   if S_{rel} then
   $S'_n \leftarrow S'_n \cup Temp$;
   else
   $S'_p \leftarrow S'_p \cup Temp$;
3. $S'_p \leftarrow S'_p - S_n$; $S'_n \leftarrow S'_n - S_p$;
4. $Dup \leftarrow S'_p \cap S'_n$; $S'_p \leftarrow S'_p - Dup$; $S'_n \leftarrow S'_n - Dup$;

Figure 1: Basic expansion function for seed sets.

("if a term belongs to S'_p (resp. S'_n), it cannot be added to S'_n (resp. S'_p)") and *Tie-break* ("if a term is added at the same time to both S'_p and S'_n, it is not useful, and can thus be eliminated from both"). The relations we have 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

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 WN relations are defined on word senses, rather than words, and since WN word senses are POS-tagged[12].

After evaluating the effectiveness of individual relations (see Section 5), we have chosen to further investigate the combination of the best-performing ones, i.e.: $Syn(J) \cup Ant^D(J)$, $Syn(J) \cap Ant^D(J)$, $Syn(J) \cup Ant^I(J)$, $Syn(J) \cap Ant^I(J)$, and the corresponding versions not restricted to adjectives.

In the experiments, we have used these relations iteratively, starting from the seed set (S_p, S_n) and producing various chains of expansion, iterating until no other terms can be added to $S'_p \cup S'_n$[13].

4.3 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[14] (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 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 WN have instead a regular format, that allows the production of cleaner textual representations (Figure 2 for an example). In WN, the senses of a word t are grouped by POS; each sense $s_i(t)$ of t is associated to (a) a list of descriptive terms that characterize $s_i(t)$[15], (b) the gloss that describes $s_i(t)$, and (c) a list of example phrases in which t occurs in the $s_i(t)$ sense. While descriptive terms and glosses usually contain terms that have a strong relation with the target term t, example phrases often do not contain any term related to t, but only t in a context of use.

We have tested four different methods for creating textual representations of terms. The first one puts together the descriptive terms and the glosses (we dub it *the DG method*), while the second also includes the sample phrases (*the DGS method*); if the lexical relation used for expansion is limited to a given POS (e.g. adjectives), we use only the glosses for the senses having that POS. We have derived the third and fourth method by applying to the DG and DGS textual representations *negation propagation* [1], that consists in replacing all the terms that occur after a negation in a sentence with negated versions of the term (e.g. in the sentence `This is not good`, the term `good` is converted to the term $\neg$`good`), thus yielding the $DG_\neg$ and $DGS_\neg$ methods.

4.4 Classification

We have classified terms by learning a classifier from the vectorial representations of the terms in (S'_p, S'_n), and by then applying the 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.

[12] In the experiments reported in this paper 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.

[13] We have reached a maximum of 16 iterations for the Ant^D relation when used on the Kam seed set.

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

[15] We have also ran some experiments in which we have used the descriptive terms directly in the expansion phase, by considering them synonyms of the target term. These experiments have not produced positive results, and are thus not reported here.

```
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 2: WordNet output for the term `unfortunate`.

The learning algorithms we have tested are the naive Bayesian learner using the multinomial model (NB), support vector machines using linear kernels, and the PrTFIDF probabilistic version of the Rocchio learner [8][16].

5. RESULTS

The various combinations of choices of seed set, 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 5 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 $DGS_{\neg}$ representations is already better than the best accuracy reported in [9] on the same term set.

Table 5 shows an average 4.4% increase (with standard deviation $\sigma = 1.14$) in accuracy in using DGS representations versus DG 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 $DGS_{\neg}$ representations. For this reason in the rest of the paper we only report results obtained used the $DGS_{\neg}$ method.

Applying expansion methods to seed sets improves results just after a few iterations. Figure 3 illustrates the accuracy values obtained in the classification of the TL term set by applying expansion functions to the *Kam* seed set, using the various lexical relations or combinations thereof listed in Section 4.2. 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`).

[16]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: 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	DG	48.47	53.41	50.01
Kam	DGS	52.47	54.63	53.69
Kam	$DG_{\neg}$	50.53	55.84	56.14
Kam	$DGS_{\neg}$	**53.81**	**58.55**	**58.76**
Tur	DG	57.86	64.49	56.91
Tur	DGS	59.56	65.10	61.06
Tur	$DG_{\neg}$	59.03	66.92	62.61
Tur	$DGS_{\neg}$	**61.18**	**68.53**	**65.49**

Figure 3 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 S'_p; 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 3 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 2, 3 and 4 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 [6, 9, 18].

On the HM test set (Table 2) the best results are obtained with SVMs (87.38% accuracy), using the *Kam* seed set and

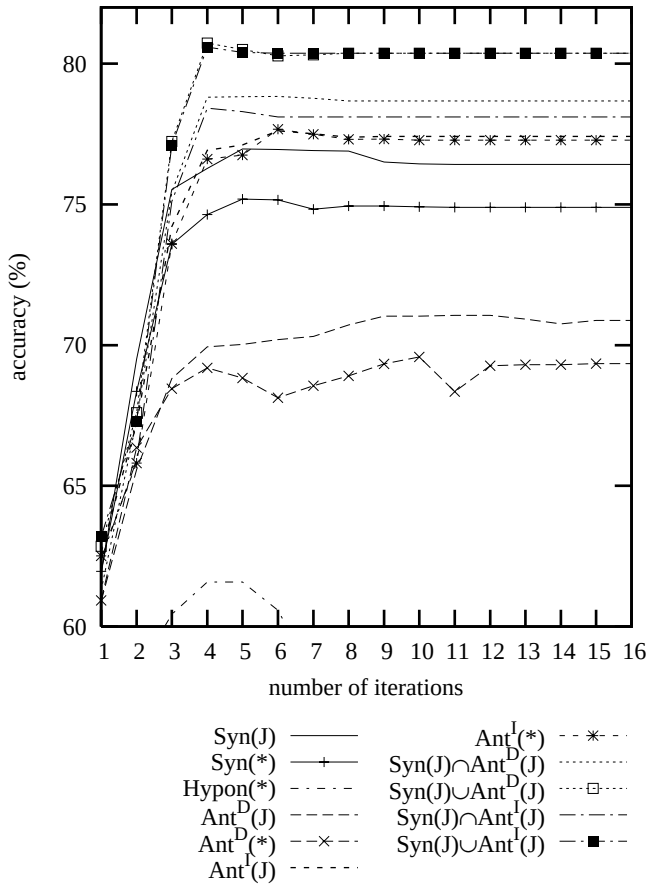

Figure 3: Accuracy in the classification (NB classifier) of the TL term set, using various lexical relations to expand the *Kam* seed set.

the $Syn(J) \cup Ant^I(J)$ relation. Our best performance is 0.3% better than the best published result [18] and 12% better than the result of [6] on this dataset.

On the TL test set (Table 3) 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 [18].

On the KA test set (Table 4) 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 [9].

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 4.3) instead of WN. 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 WN.

In our comparisons with previously published methods we note that, while improvements with respect to the methods of [6, 9] have been dramatic, the improvements with respect to the method of [18] have been marginal. However, compared to the method of [18], ours is much less data-intensive:

Table 2: Best results in classification of HM.

Method	Seed set	Expansion method	# of iterations	Acc. (%)
[6]	–	–	–	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
[18]	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 3: Best results in the classification of TL.

Method	Seed set	Expansion method	# of iterations	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
[18]	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

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 [18] (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 [18] required about 70 hours. In an experiment using a volume of data only 20 times the size of ours (10 million word occurrences), [18] 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 [18], actually a subset of it containing only 1+1 seed terms instead of 7+7[17].

6. CONCLUSIONS

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 [18]. This result is valuable 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 [18]

[17]Additionally, we should mention that our results are also fully reproducible. This is not true of the results of [18], 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.

Table 4: Best results in the classification of KA.

Method	Seed set	Expansion method	# of iterations	Acc. (%)
[9]	*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

7. ACKNOWLEDGMENTS

This work was partially supported by Project ONTOTEXT "From Text to Knowledge for the Semantic Web", funded by the Provincia Autonoma di Trento under the 2004–2006 "Fondo Unico per la Ricerca" funding scheme.

8. REFERENCES

[1] S. R. Das and M. Y. Chen. Yahoo! for Amazon: Sentiment parsing from small talk on the Web. In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*, Barcelona, ES, 2001.

[2] K. Dave, S. Lawrence, and D. 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, New York, US.

[3] S. D. Durbin, J. N. Richter, and D. Warner. A system for affective rating of texts. In *Proceedings of OTC-03, 3rd Workshop on Operational Text Classification*, Washington, US, 2003.

[4] Z. Fei, J. Liu, and G. Wu. Sentiment classification using phrase patterns. In *Proceedings of CIT-04, 4th International Conference on Computer and Information Technology*, pages 1147–1152, Wuhan, CN, 2004.

[5] G. Grefenstette, Y. Qu, J. G. Shanahan, and D. A. Evans. Coupling niche browsers and affect analysis for an opinion mining application. In *Proceedings of RIAO-04, 7th International Conference on "Recherche d'Information Assistée par Ordinateur"*, pages 186–194, Avignon, FR, 2004.

[6] V. Hatzivassiloglou and K. 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.

[7] V. Hatzivassiloglou and J. M. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of COLING-00, 18th International Conference on Computational Linguistics*, pages 174–181, 2000.

[8] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In D. 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.

[9] J. Kamps, M. Marx, R. J. Mokken, and M. D. 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.

[10] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of COLING-04, 20th International Conference on Computational Linguistics*, pages 1367–1373, Geneva, CH, 2004.

[11] S. Morinaga, K. Yamanishi, K. Tateishi, and T. 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.

[12] T. Nasukawa and J. 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.

[13] B. Pang and L. 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.

[14] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP-02, 7th Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, US, 2002. Association for Computational Linguistics, Morristown, US.

[15] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. In W. Daelemans and M. Osborne, editors, *Proceedings of CONLL-03, 7th Conference on Natural Language Learning*, pages 25–32, Edmonton, CA, 2003.

[16] P. J. Stone, D. C. Dunphy, M. S. Smith, and D. M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, US, 1966.

[17] P. 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, 2002.

[18] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346, 2003.

[19] T. Wilson, J. Wiebe, and R. 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*, San Jose, US, 2004.

[20] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In M. Collins and M. Steedman, editors, *Proceedings of EMNLP-03, 8th Conference on Empirical Methods in Natural Language Processing*, pages 129–136, 2003.