# Extracting Opinions and Facts for Business Intelligence

Horacio Saggion*, Adam Funk*

*Department of Computer Science
University of Sheffield
Regent Court
211 Portobello Street
Sheffield - S1 5DP
{H.Saggion,A.Funk}@dcs.shef.ac.uk

**Abstract.** Finding information about companies on multiple sources on the Web has become increasingly important for business analysts. In particular, since the emergence of the Web 2.0, opinions about companies and their services or products need to be found and distilled in order to create an accurate picture of a business entity. Without appropriate text mining tools, company analysts would have to read hundreds of textual reports, newspaper articles, forums' postings and manually dig out factual as well as subjective information. This paper describes a series of experiments to assess the value of a number of lexical, morpho-syntactic, and sentiment-based features derived from linguistic processing and from an existing lexical database for the classification of evaluative texts. The paper describes experiments carried out with two different web sources: one source contains positive and negative opinions while the other contains fine grain classifications in a 5-point qualitative scale. The results obtain are positive and in line with current research in the area. Our aim is to use the result of classification in a practical application that will combine factual and opinionated information in order to create the reputation of a business entity.

## 1 Introduction

It is accepted that incidents which damage a company's reputation for honesty or safety may cause serious damage to finances. For example, in 1999 Coca Cola lost $60 million (by its own estimate) after schoolchildren reported suffering from symptoms like headaches, nausea and shivering after drinking its products. The reputation of a company not only depends on the quality of its products but also on the behaviours of their employees and in particular their managers. Many businesses have public relations departments dedicated to managing their reputation. The public relations industry is growing due to the demand for companies to build corporate credibility and hence reputation. One important component of the reputation of a company is the opinion that stake-holders have about it, its products, and its services. The Web has become an increasingly important source of information in all areas of society in particular, in the field of business intelligence, business analysts are turning their eyes on the web in order to obtain factual as well as more subtle and subjective information (opinions) on companies.

However, tracking what is being said about a company is not trivial; without appropriate text mining tools, company analysts would have to read hundreds of textual reports, newspaper articles, forums' postings and manually dig out factual as well as subjective information. Work on extracting factual information is related to the field of information extraction, the process of extracting from text specific facts in a given target domain (Grishman, 1997). The field of information extraction has been fuelled by two major US international evaluations efforts, from 1987 until 1997 the Message Understanding Conferences (Grishman and Sundheim, 1996) and since 2000 the Automatic Content Extraction Evaluation. Work on opinion mining has recently emerged thanks to evaluation programs such as the Text Retrieval Conference (TREC) 2006 track on blog mining for opinion retrieval or the Text Analysis Conference[1] (TAC) with a track on opinion mining and summarisation.

Opinion mining consists of several different problems, such as determining whether each segment of text (sentence, paragraph, or section) is "opinionated" or not; identifying the opinion-holder (the person or organization who expresses the opinion)[2]; determining the polarity of the opinion (how positive or negative each opinion is); and its theme or subject; for business intelligence, it is also useful to classify each opinion according to the aspect of the business or transaction described (such as service, product quality, ordering, or integrity).

Opinion analysis helps to assess the limitations of particular products and then exploit this information in the development of improved products or services. It also helps enterprises understanding their customers as well as plan for future products and services.

Given the abundance of product reviews on the World Wide Web, especially with the more recent proliferation of blogs and other Web 2.0 services, one application of opinion mining is for each given entity (e.g., product) to identify its features (e.g., size, colour) and what is being said about them (positive or negative statements). These opinions can be combined and used to produce a textual summary together with statistics about what has been said about the entity as a whole or about each of its attributes or features. Opinion summaries are useful instruments in competitive intelligence for example, because they help assess the limitations of particular products and then exploit this information in the development of improved produces or services by the producer or its competitors.

The work presented here is being carried out in the context of the MUSING project, in which we are applying human language technology in a process of ontology-based extraction and population in the context of business intelligence applications (Saggion et al., 2007). Business intelligence (BI) is the process of finding, gathering, aggregating, and analysing information to support decision-making. It has become evident to business analysts that *qualitative* information (as opposed to quantitative data such as share prices) plays an important role in many BI applications. One such application in MUSING is a reputation teller that aims to collect and organise opinions about business entities (organisations, people, products, etc.). In MUSING, information is organised in a domain ontology, which the information extraction systems target. In particular, a sub-ontology in MUSING models subjective information such as reputation, reliability, and quality. The reputation teller's overall objective is to identify statements which reflect these concepts and track them over time in order to create an accurate picture of a business entity. Each company, person, etc., extracted by our system is stored with its instantiated properties in a knowledge repository based on an ontology of the application

[1] `http://www.nist.gov/tac/`

[2] This can also be treated as an information extraction problem (Riloff et al., 2002).

domain. (The ontology population process has been reported elsewhere (Yankova et al., 2008) and is not repeated here.) The repository includes qualitative information associated with the entities; thus all information is integrated.

Here we present extensive work on the use of natural language processing to contribute to the reputation teller application, which targets both factual and opinionated discourse. In particular, we aim in the experiments described below to establish the reliability and utility of lexical, morphological, syntactic, and sentiment-based features for the identification of opinions in text.

The paper is organised as follows: In Section 2 we discuss a practical business intelligence application requiring the identification and classification of opinions; next in Section 3 we present related work on sentiment analysis. In Section 4, we introduce our linguistic analysis technology and machine learning framework. In Section 5 we describe our opinion mining framework and in Sections 6 and 7 we present our experimental results. Finally, Section 8 discusses our results in relation to state of the art and Section 9 concludes the paper.

# 2 A Business Intelligence Application

The objective of a reputation teller is to track the reputation of a given company in a period of time. While traditionally, the reputation of a company is computed based on financial indicators, the industry is keen on including qualitative information in the association of reputation indices to companies. In this sense, the reputation of a company can be thought as *an index of what the community thinks about the company*. In order to capture what is being said about a company (positive, negative, neutral statements), business analysts need to gather and analyse textual sources. Two important sources of information can be used in gathering company information: Web pages of the companies themselves, providing useful factual information about products, services, executives, revenues, etc. and public forums where the common men can express their opinions. The schema of the application is shown in Figure 1.

Figure 1 also shows the data-sources used for mining factual and subjective information about companies. For factual information, we use a process to crawl pages from company web sites of interest. For subjective information, we have crawled web pages on two fora.

- From one consumer forum[3], we have collected a corpus of HTML documents, each containing in particular a comment (a paragraph of natural-language text) and a *thumbs-up* or *thumbs-down* rating, both entered by one of the forum's users. Each rating was represented by an `<img>` tag pointing to a GIF cartoon of a thumbs-up or thumbs-down gesture, with an `alt` attribute of `Consumer ThumbsUp` or `Consumer ThumbsDown`, respectively. The crawling process starts with some seed pages which are used to explore all available postings. See Figure 2 for an example.

- From another consumer forum[4], we have collected a corpus of HTML pages, each containing a number of separate comments product or company reviews. Each review consisted of a paragraph or two of natural-language text entered by one of the forum's users and the same user's rating of the company from one to five stars. Each rating was represented by an `<img>` tag pointing to a GIF image of a row of one to five adjacent stars,

[3] `http://www.clik2complaints.co.uk/`
[4] `http://www.pricegrabber.co.uk/`

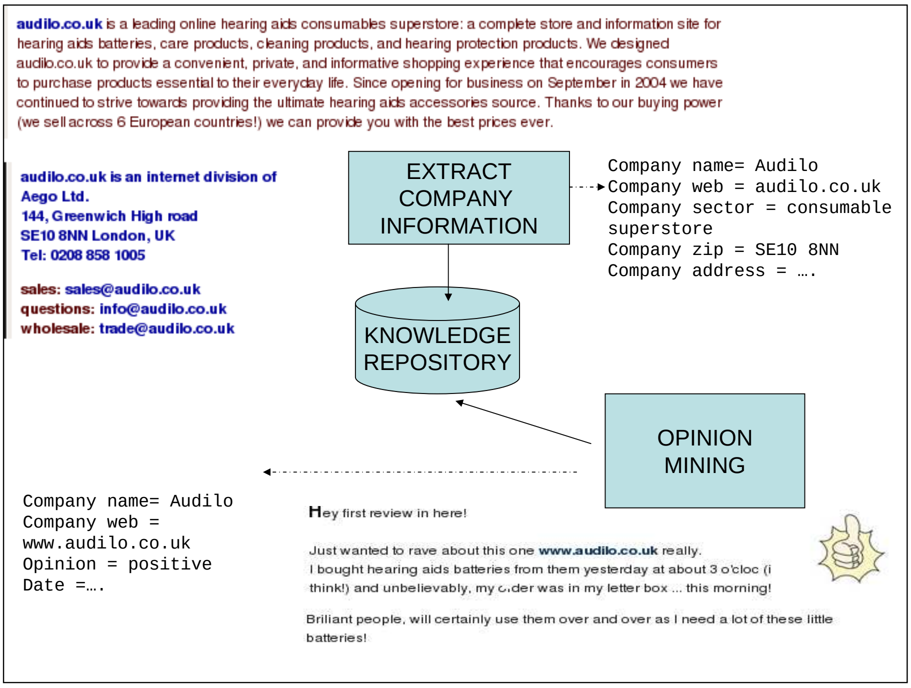


FIG. 1 – *Data Sources for Company Reputation. The top of the figure shows one company web page which feeds a process of ontology-based information extraction. The bottom of the figure shows data sources used to mine opinions about the company.*

with an `alt` attribute of `1 Star Review`, `2 Star Review`, etc. Here as well only a few seed pages are used to extract all reviews from the forum. See Figure 3 for an example.

Subjective statements about a given company are *merged with factual data extracted for that company by the company extraction module thanks to a process of identity resolution*. In the example in Figure 1, the company extraction module will identify the name and Internet address of the company which will be matched against the Internet address and name extracted from the review, we have used these two sources of information together with address information as relevant features in our ontology population and merging procedure (Yankova et al., 2008).

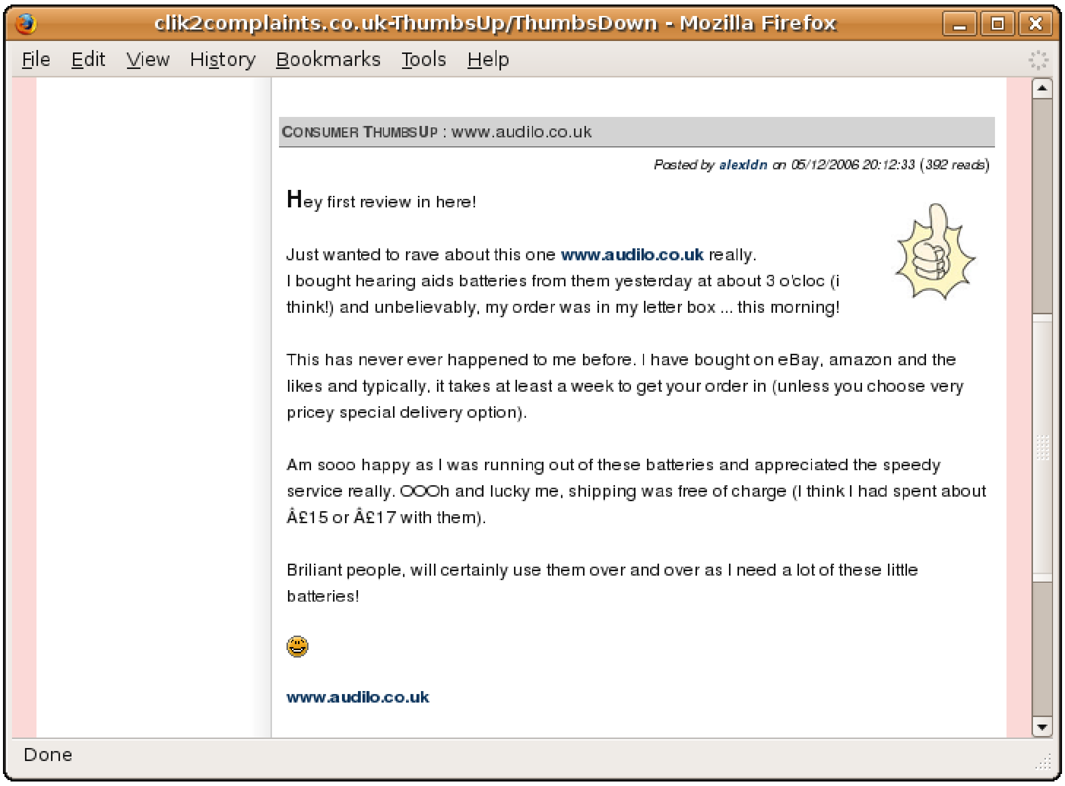


FIG. 2 – *Positive/Negative reviews.*

# 3 Related work

Classifying product reviews is a common problem in opinion mining: the goal is to identify for a given entity its features and the positive or negative statements expressed then identify what is being said about each of them. This information is then compiled in order to produce textual summaries together with statistics about the frequency of positive, negative, and neutral statements. A variety of techniques have been used here including supervised (Li et al., 2007a) and unsupervised (Hu and Liu, 2004; Turney, 2002; Zagibalov and Carroll, 2008; Zhuang et al., 2006) machine-learning.

Language resources such as SentiWordNet have recently been developed for the research community (Esuli and Sebastiani, 2006). Some approaches to opinion mining involve predefined gazetteers of positive and negative "opinion words", whereas Turney's well-known method (Turney, 2002) determined the semantic orientation of lexemes by calculating their Pointwise Mutual Information (PMI, based on probability of collocations (Church and Hanks, 1990)) to the reference words *excellent* and *poor*. More recent work on product reviews in particular involved the identification of words referring to *implicit* and *explicit features* (Liu et al., 2005). Naturally, the work based on unsupervised learning has relied on *a priori* information.

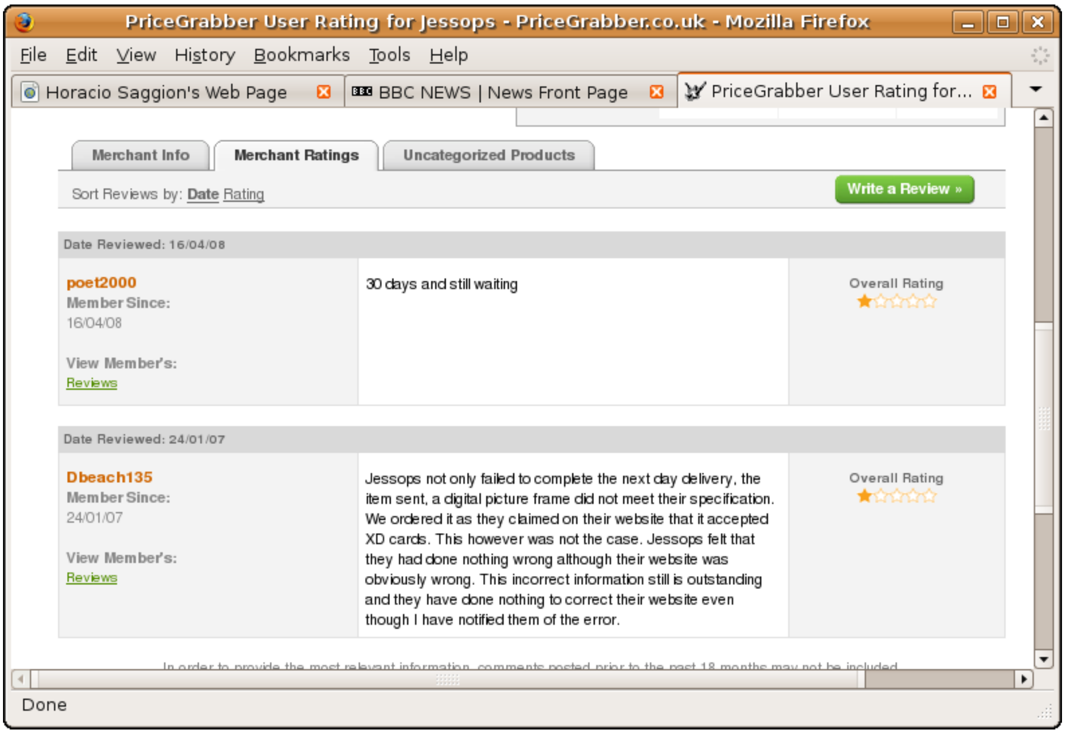


FIG. 3 – *Fine grained reviews (1-5 stars).*

Devitt and Ahmad (2007) presented work similar to ours in the sense that they also deal with the business domain. They are interested in two problems related to financial news: identifying the polarity of a piece of news, and classifying a text in a fine 7-point scale (from very positive to very negative). They propose a baseline classifier for positive/negative distinction which has an accuracy of 46% and have more sophisticated classifiers based on lexical cohesion and SentiWordNet achieving a maximum of 55% accuracy.

Aue and Gamon (2005) combine a semantic orientation method based on Turney's pairwise mutual information approach with an approach based on the assumption that terms with opposite orientation tend not to co-occur at the sentence level (maybe in contradiction with the assumption by Hatzivassiloglou and McKeown (1997) that this can occur depending on particular syntactic contexts—"interesting and useful" versus "beautiful but boring"). They test this idea in a classification task—which consist on classifying sentences into positive, neutral, and negative—achieving around 50% accuracy.

Dave et al. (2003) presents several techniques to create features (words or terms) and associated scores from training corpora for a classification task which consist on sifting positive and negative statements associated to product reviews from *Cnet reviews* and *Amazon*. They investigate various mechanisms to produce features—with a unigram model as baseline and more complex models employing lexical substitution, higher n-grams, and syntactic phrases—

and weighting mechanisms such as inverted document frequency. Their classifier aggregates features' scores for sentences and bases the classification on the sign of the aggregated score. The use of simple n-grams seem to perform better than any other investigated feature generation technique and n-grams grater that one seem to perform better than unigrams. The proposed technique achieves over 80% classification accuracy.

Ghose et al. (2007) investigate the issue of generating in an objective way a lexicon of expressions for positive and negative opinion. They note that expressions such as "good" can be considered not so positive but quite negative in some contexts such as in e-commerce. They investigate the correlation of monetary gain with the occurrence of particular phrases in merchants' "reputation profiles". The reputation profiles contain reviews of the services offered by the merchants and qualify characteristics of the merchant such as "delivery", "packaging", and overall "service". By correlating gain with reviews they obtain an objective ranking of phrases which influence the monetary gain a merchant can make.

Our work aims to identify how lexical semantic information can be used together with a data-driven approach based on language processing (NLP) techniques as input to a machine learning tool.

## 4 Text Processing: Natural Language Processing (NLP) Tools

In this paper, linguistic analysis of textual input is carried out using the General Architecture for Text Engineering (GATE). GATE is a framework for the development and deployment of language processing technology in large scale (Cunningham et al., 2002). GATE is free software hosted on SourceForge[5]). It provides three types of resources for developing NLP applications: Language Resources (LRs) which contain data; Processing Resources (PRs) which contain algorithms; and Visualisation Resources (VRs) which represent visualisation and editing components. GATE can be used to process documents in different formats including plain text, HTML, XML, RTF, and SGML. Textual input is transformed with the GATE software in a GATE document: a LR which will contain the input text together with one or more sets of annotations (one of which will represent the document markups if any). Annotations are generally updated by algorithms manipulating the document (PRs) during text analysis. Each annotation in a GATE document belongs to an annotation set and has a type, a pair of offsets (the span of text one wants to annotate), and a set of features and values that are used to encode the information. Features (or attribute names) are strings, and values can be any Java object. Attributes and values can be specified in an annotation schema which facilitates validation and input during manual annotation. Programmatic access to the annotation sets, annotations, features and values is possible through the GATE Application Program Interface.

Various processing resources available in GATE are used in our work including:

- a tokeniser, which segments the text of the document in units representing words, punctuation, and other elements. GATE produces a token annotation for each word in the document. Tokens' features computed during this process are the type of tokens (word, punctuation, number, space, control character, etc.), their lengths, and their orthographic characteristics (all capitals, all lowercase, capital initial, etc);

[5] `http://sourceforge.net/projects/gate`

- a sentence splitter which segments the text into sentences;
- a parts of speech tagger: This is the process of associating to each word form or symbol a tag representing its part of speech. In GATE, it is implemented with a modified version of the Brill tagger (Brill, 1992).
- a morphological analyser: with decompose each word into its root (or lemma) and affixes.
- a semantic annotation process: This process consists on the recognition and classification of a set of entities in the document, commonly referred as to named entity (NE) recognition task. NE recognition is a key enabler of information extraction the identification and extraction of key facts from text in specific domains. Two approaches to NE recognition are possible in GATE. A manually developed set of rules based on corpus analysis or a supervised or semi-supervised approach using machine learning tools and features designed by human experts. We are applying both in this work.
- a co-reference resolution process to identify equivalent names in text;
- a noun chunker which identifies basic (i.e., non-recursive) noun phases; and
- a wrapper for the Stanford parser[6], which produces dependency structures (Tesnière, 1959) for each sentence in the document.

All components described here are being used in the information extraction and opinion mining application we describe in this paper. In particular, named entity recognition and co-reference resolution are applied to identify entities in text, and noun chunking and parsing are used to extract phrases which are associated to entities targeted in the application. A detail of components used for the opinion mining application is given in Section 5.2.

## 4.1 Machine Learning Tools for Opinion Mining

Statistical machine learning approaches to information extraction include the use of Hidden Markov Models (HMM), Support Vector Machines (SVM), and Conditional random Fields (CRF). With HMMs (Leek, 1997) the information extraction task is cast as a tagging problem where, given a sequence of input words, the system has to produce a sequence of tags; the words are observations and the tags are hidden states in the HMM. CRFs (Lafferty et al., 2001) are state-of-the-art techniques for IE and tend to do better than other classification methods. SVMs are very competitive supervised models for information extraction (Isozaki and Kazawa, 2002), which treat the task as a binary classification problem (or set of intersecting binary problems; each label gives rise to a binary classification problem) by seeking to optimise a hyperplane in the vector space of instances that maximally separates positive from negative instances. SVMs have been used in a variety of NLP problems which are instances of multi-class classification problems (for more than two classes; in named entity recognition, for example, there are a considerable number of names to be recognised such as location names, organisation names, personal names) and perform well in this field (Li et al., 2005, 2007b). We

[6] `http://nlp.stanford.edu/software/lex-parser.shtml`

adopt SVM learning paradigm not only because it has recently been used with success in different tasks in natural language processing, but it has been shown particularly suitable for text categorisation (Joachims, 1998). (In previous classification experiments, we have tried other machine learning algorithms such as Decision Trees, Naive Bayes Classification, and Nearest Neighbour from the Weka toolkit (Witten and Frank (1999)), but support vector machines[7] gave us best overall classification accuracy.)

Almost all these statistical approaches adopt the same steps: first they transform the problem into a multi-class classification task; they then convert the multi-class problem into several binary classification problems using a one-vs-all or one-vs-another approach (for example); then an SVM classifier is trained for each binary classification task; finally, the classifiers' results are combined to obtain the solution to the original NLP problem.

In our methodology each information extraction learning problem is transformed into a classification problem. Each learning instance is transformed into a vector representation in a high dimensional feature space (we use lexical, syntactic, and semantic features). The SVM learns a hyperplane that separates positive from negative instances with the maximal distance to all training examples. This work uses SVM with uneven margin as proposed by Li and Shawe-Taylor (2003).

A binary SVM classifier corresponds to a hyperplane in feature space with maximal margin, which separates the positive and negative training examples. The margin can be regarded as a measure of the error-tolerance ability of the classifier, since a classifier is more likely to classify a test instance correctly if it has a larger margin. In general, if a training set is representative of the whole dataset, a classifier with a larger margin with respect to the training set would have a better generalisation performance. However, if the training set is unrepresentative, then a maximal margin classifier (such as SVM) learnt from an unrepresentative training set may have poor generalisation performance. Many imbalanced classification problems, such as those arising in information extraction, have only a small number of positive training examples, resulting in an SVM classifier with poor generalisation capability (only a few tokens in token-based classification are positive while most tokens do not belong to any target concept). If an SVM classifier has to be learnt from an imbalanced training set which has only a few positive examples, it may be beneficial to require the learning algorithm to set the margin with respect to the positive examples (the positive margin) to be somewhat larger than the margin with respect to the negative examples (the negative margin). In other words, in order to achieve better generalisation performance, one needs to distinguish the positive margin from the negative margin when training the SVM. A margin parameter is introduced into the SVM optimisation problem to control the ratio of the positive margin over the negative margin (for details see Li and Shawe-Taylor (2003)). Please see Li et al. (2009) for fuller mathematical details of the SVM with uneven margin (SVMUM) technique.

When applying SVMUM to a problem, we need to identify the value for the uneven margins. If the problem has just few positive training examples and many negative ones, then a margin smaller than 1 could be used. The margin parameter can be empirically determined by cross-validation on training data. A reasonable estimation of the margin parameter can help achieve better performance than using a standard SVM. Some problems or data sets may not be sensitive to changes in the margin, thus an standard SVM can be applied. A second parameter

[7]We used the SVM support vector machines adopted by the GATE platform.

which has to be carefully selected in the SVM algorithms is the probability threshold (between 0 and 1) to be used to accept or reject a particular classification.

In order to estimate these parameters empirically for the classifiers we will present, we use a set of $n$ documents from the corpus and carry out an experiment for each possible combination of probability and margin using values between 0.10 and 1.00 with steps of 0.10. For each pair of values, $n$ iterations are executed where document $i$ is removed from the corpus, the $n-1$ documents remaining documents are used for training the SVM with the given parameters, and the $i$ document is used to test the algorithm. At each iteration precision, recall, and f-score are computed. The probability and margin are chosen as the ones maximising the f-score.

## 4.2 Extracting Company Information

Extracting company information consist on the identification of pieces of information about a company modelled in an ontology of the domain. We use information extraction techniques to transform unstructured and semi-structured documents into structured representations—RDF statements for ontology population. The concepts targeted by this application are the company name, its main activities, its number of employees, its board of directors, turnover, etc. (full list of concepts is shown in Table 1).

| **Concept in the ontology** | **Meaning** |
|---|---|
| Company Name | full name of the company and aliases |
| Address | including Street, Town, Country, Postcode etc |
| Email Address | any e-mail |
| Fax number | all fax numbers |
| Telephone number | all telephone numbers |
| Website | the url of the company web site |
| Industry Sector | the industrial sector the company belongs to |
| Number of Employees | how many employees the company has in total |
| Establishment Date | the date the company was first established |
| Activity Area | the main activity area of the company |
| Product | the products produced by the company |
| Services | the services produced by the company |
| Market Outlet | the main market outlet for the company |
| Award | the awards the company has won |
| Executive | the names and roles of the company executives |
| Partnership | any business partners the company has |

TAB. 1 – *List of Extracted Concepts.*

The information extraction system has been developed with the GATE platform and the component previously described. The extraction prototype uses some default linguistic processors from GATE, but the core of the system (the concept identification program) was developed specifically for this application. In addition to specific processes such as phrase chunking, lexicons and gazetteer lists have been created to perform gazetteer lookup processes. Rules for concept identification have been specified in regular grammars implemented in the JAPE language. JAPE is part of GATE and is used to write regular expressions over annotations which are used to identify word sequences as belonging to specific semantic categories (e.g., organization names, person names, measurements, dates, etc.). The result of the automatic annotation is further analysed by (i) a module which produces RDF triples associating different pieces of information together (e.g., a company with its number of employees, a company with its CEO), and (ii) the ontology population module responsible for knowledge base population.

To evaluate the extraction system, we use traditional metrics for information extraction (Chinchor, 1992): precision, recall, and F-measure. Precision measures the number of correctly identified items as a percentage of the number of items identified. It measures how many of the items that the system identified were actually correct, regardless of whether it also failed to retrieve correct items. The higher the precision, the better the system is at ensuring that what is identified is correct. Recall measures the number of correctly identified items as a percentage of the total number of correct items measuring how many of the items that should have been identified actually were identified. The higher the recall rate, the better the system is at not missing correct items. The F-measure (van Rijsbergen, 1979) is often used in conjunction with Precision and Recall, as a weighted average of the two—usually an application requires a balance between Precision and Recall. An evaluation of the performance of the extraction system indicates good results with over 84% F-score. It is worth noting that we are also exploring the use of machine learning techniques for information extraction and we have obtained a similar performance when applying SVM to the identification of company information. We use 10-fold or 5-fold cross-validation in order to make the most thorough evaluation over our corpora.

# 5 Opinion Mining

In this paper we are dealing with the problem of classification of opinionated texts (e.g., review) in positive/negative or in a fine grain classification scale (e.g., very bad to excellent). Because we have access to considerable free annotated training data, we solve the classification problem in a supervised machine learning framework. Our objective is to use the classification system to recognise positive and negative opinions over data sources which have not been annotated. We also plan to use the classifiers to filter out opinionated vs non-opinionated sentences. Finally, the extraction tools described before are being used to identify the object of the opinion (e.g., company) and the particular characteristic being criticised or praised (e.g., product, service). In our learning framework each text represent a learning or testing instance.

## 5.1 Instances for Learning

Each learning instance is represented as a vector of feature-values, in our case features are created from linguistic annotations produced by different linguistic processors. The features to be used are selected according to hypothesis one may have about what may influence recognition of a class. In the case of sentence or text classification which is the focus of the work presented here, the features are either lexical (morphological information), syntactic (relying on parts of speech information), semantic (relying on a sentiment dictionary), and discursive (textual).

### 5.1.1 SentiWordNet

SentiWordNet (Esuli and Sebastiani, 2006) is a lexical resource in which each synset (set of synonyms) of WordNet (Fellbaum, 1998) is associated with three numerical scores $obj$ (how objective the word is), $pos$ (how positive the word is), and $neg$ (how negative the word is). Each of the scores ranges from 0 to 1, and their sum equals 1. SentiWordNet word values have been semi-automatically computed based on the use of weakly supervised classification algorithms. Examples of "subjectivity" scores associated to WordNet entries are shown in Table 2, the entries contain the parts of speech category of the displayed entry, its positivity, its negativity, and the list of synonyms. We show various synsets related to the words "good" and "bad". There are 4 senses of the noun "good", 21 senses of the adjective "good", and 2 senses of the adverb "good" in WordNet. There is one sense of the noun "bad", 14 senses of the adjective "bad", and 2 senses of the adverb "bad" in WordNet.

| Category | WNT Number | pos | neg | synonyms |
|---|---|---|---|---|
| a | 1006645 | 0.25 | 0.375 | good#a#15 well#a#2 |
| a | 1023448 | 0.375 | 0.5 | good#a#23 unspoilt#a#1 unspoiled#a#1 |
| a | 1073446 | 0.625 | 0.0 | good#a#22 |
| a | 1024262 | 0.0 | 1.0 | spoilt#a#2 spoiled#a#3 bad#a#4 |
| a | 1047353 | 0.0 | 0.875 | defective#a#3 bad#a#14 |
| a | 1074681 | 0.0 | 0.875 | bad#a#13 forged#a#1 |

TAB. 2 – *Examples of SentiWordNet Entries. "good#15" means sense number 15 of the word "good".*

In order to identify the positivity or negativity of a given word in text, one first needs to perform general word sense disambiguation, i.e., when observing a word such as "good" in text, and assuming it is an adjective, one would have to decide for one of its 21 senses. Instead, we do not apply any word sense disambiguation procedure: for each entry in SentiWordNet (each word#sense) we compute the number of times the entry is more positive than negative (positive > negative), the number of times is more negative than positive (positive < negative) and the total number of entries word#sense in SentiWordNet, therefore we can consider the overall positivity or negativity a particular word has in the lexical resource. We are interested in words that are generally "positive", generally "negative" or generally "neutral" (not much variation between positive and negative). For example a word such as "good" has many more entries where the positive score is greater than the negativity score while a word such as "unhelpful" has more negative occurrences than positive. We use this aggregated scores in our experiments on opinion identification. A language resource has been implemented in GATE to access the SentiWordNet resource and an algorithm to compute the "general" sentiment of a word has been implemented.

## 5.2 Linguistic and Semantic Features

Here we describe the features we use in this paper to represent instances. For each token in the document the following features are used in our experiments:

- *string* the original, unmodified text of the token;
- *root* the lemmatised, lower-case form of the token (for example, *run* is the root feature for *run*, *runs*, *ran*, and *Running*);
- *category* the part-of-speech (POS) tag, a symbol that represents a grammatical category such as determiner, present-tense verb, past-tense verb, singular noun, etc.)[8];

[8]Our POS tagger uses the Wall Street Journal corpus's tagset.

- *orth* a code representing the token's combination of upper- and lower-case letters[9] (if it has been classified as a word).
- *countP* the word's positivity score (base on our description);
- *countN* the word's negativity score;
- *countF* the total number of entries for the word in SentiWordNet.

We additionally use the following syntactic features (*syn_features*):

- *ADJ* the lemmatized form of an adjective;
- *ADV* the lemmatized form of an adverb;
- *ADJ_ADJ* a bigram of adjectives' lemmas;
- *ADV_ADV* a bigram of adverbs' lemmas;
- *ADV_ADJ* a bigram of adjective's lemma and adverb's lemma.

For each sentence in the document the following features are used (*sentence_features*):

- *countP* (at sentence level) the number of positive words in the sentence (words which have been observed with a positive polarity more[10] times than with a negative polarity);
- *countN* (at sentence level) the number of negative words in the sentence (words which have been observed with a negative polarity more[11] times than with a positive polarity);
- *senti* a value 'pos' or 'neg' or 'neutral' according to the distribution of sentiP and and sentiN in the sentence[12].

For each target text fragment in the document the following features are used (*text_features*):

- *count_pos* the number of sentences with *senti* value 'pos';
- *count_neg* the number of sentence with *senti* value 'neg';
- *count_neutral* the number of sentences with *senti* value 'neutral'.

All these features are computed by specially designed programs. In Figures 4 to 6, we show the partial result of the linguistic and semantic analysis of the documents.

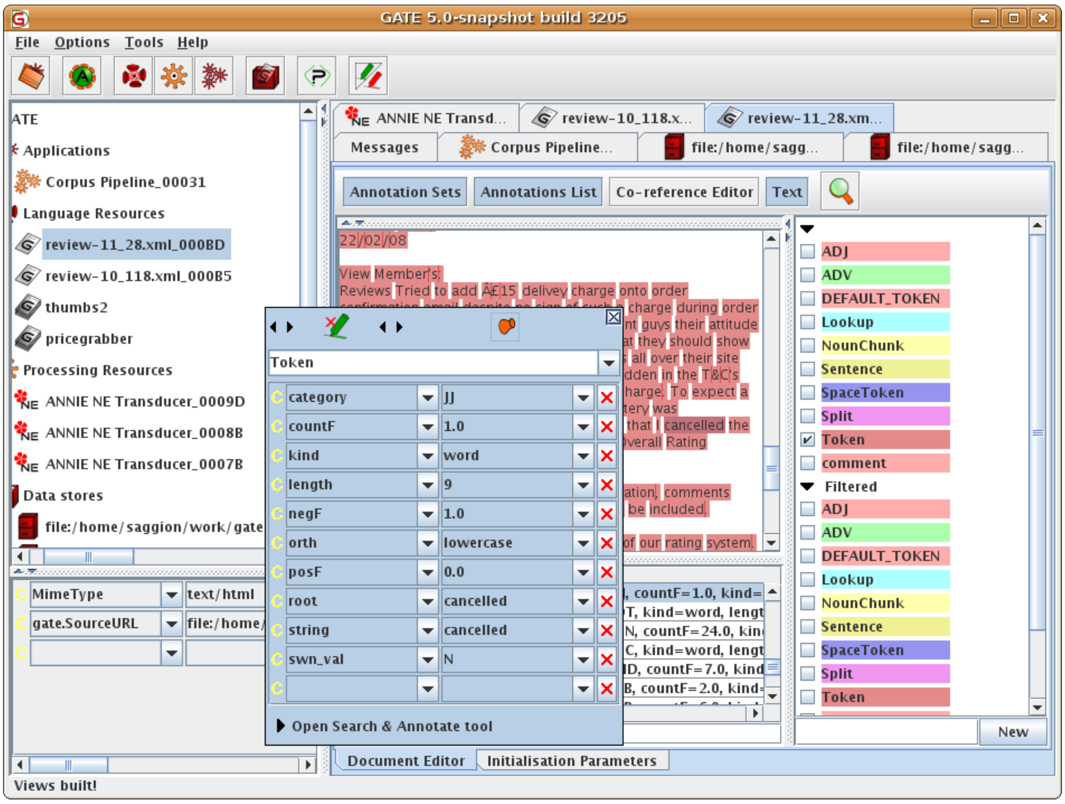


FIG. 4 – *Annotation of a word with linguistic and semantic information from SentiWordNet.*

In Figure 4 we show the features which have been produced for the token "cancelled" which is an adjective (e.g., parts-of-speech category "JJ"). The positivity and negativity scores and a feature indicating our computation of the SentiWordNet value (snw_val, which in this case is negative) are shown. In Figure 5, we show the counts for positive and negative words and the "senti" feature which indicates whether the sentence is positive, negative, or neutral. Finally, Figure 6, shows the features computed for a review, based on the aggregation of features for sentences and words in the review. The sentence also contains a feature to represent the "true" classification (e.g., "rating").

[9] `upperInitial`, `allCaps`, `lowerCase`, or `mixedCaps`

[10] The positive score (countP) is greater than half the total number of entries of the word in SentiWordNet (countF)

[11] The negative score (countN) is greater than half the total number of entries of the word in SentiWordNet (countF)

[12] 'pos' will be assigned when sentiP accounts for the majority of the cases; 'neg' will be assigned when sentiN accounts for the majority of the cases; 'neutral' is the default case

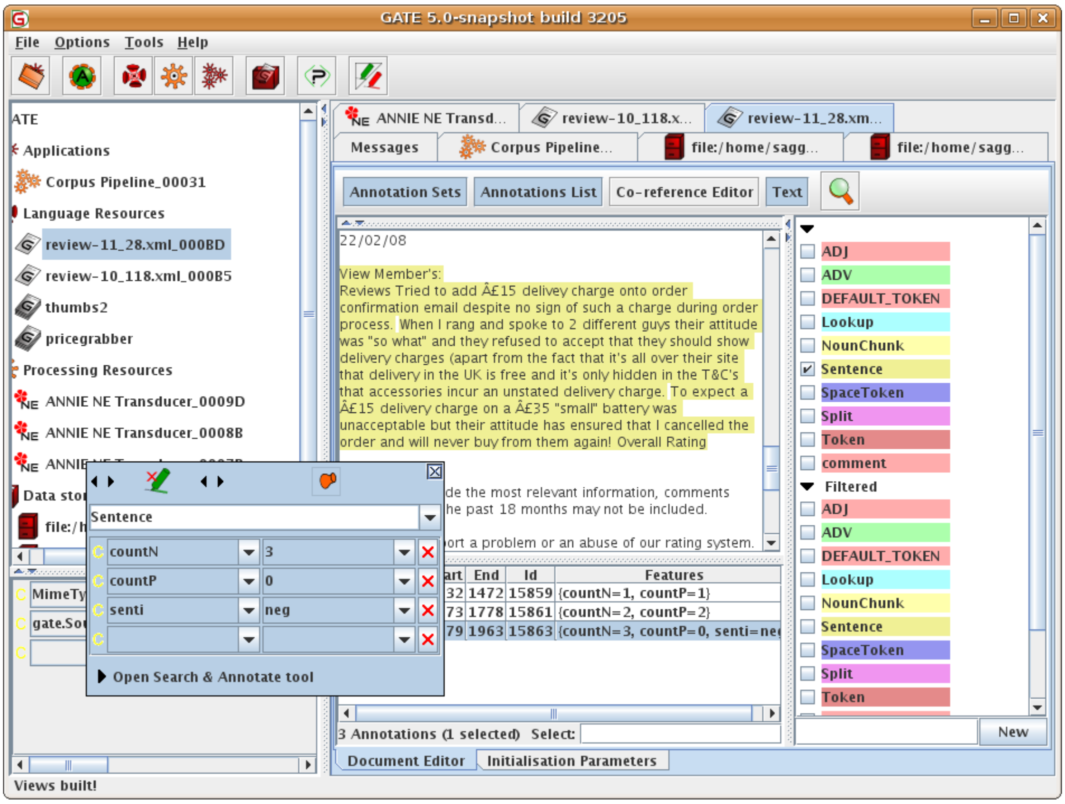


FIG. 5 – *Annotation of a sentence with information aggregated from the tokens.*

## 6 Binary Classification Experiments

The corpus of documents we are using for these experiments consisted of 92 documents, each containing one instance (review) for classification (see Table 3). The distribution of ratings in the corpus was 67% *thumbs-down* and 33% *thumbs-up*. So classifying each text as *thumbs-down* would give a classification accuracy of 67%.

Two classifiers have been used. One of the classifiers is a unigram based classifier which uses parts of speech and morphology information (*Lexical-based Classifier*). This classifier has given us very good performance in this corpus. The second set of experiments uses a more sophisticated set of features based on syntactic and semantic information. More specifically, the second classifier (*Sentiment-based Classifier*) uses the set of features presented in Section 5.2: *syn_features*, *sentence_features*, and *text_features*. As it will be shown while no differences are observed in classification accuracy some interesting features emerge for the sentiment analysis classification task. We have carried out training and evaluation within a 10-fold cross-validation framework over the *thumbs-up/thumbs-down* corpus, in order to classify each review text as *thumbs-up* or *thumbs-down* based on SVM analysis of various combinations of

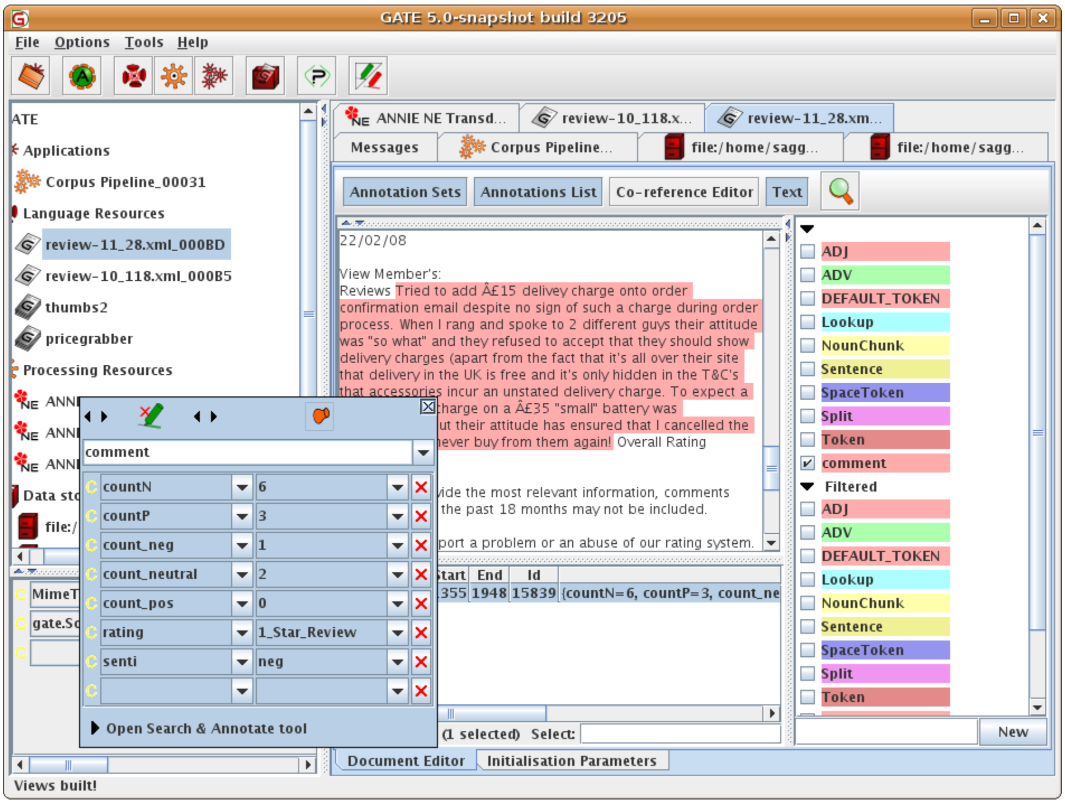

FIG. 6 – *Annotation of a review with information aggregated from the sentences.*

features listed in Section 5.2.

## 6.1 Lexical-based Classifier Results

Table 5 summarizes the standard classification measurements (averaged over 10 runs) from this series of experiments. From these results we can make the following general observations.

- The combination of *category* and *orth* produced relatively poor results—as expected, because it is semantically empty.
- Increasing the number of features does not necessarily improve performance, because it can make the training data sparse.
- Increasing the value of $n$ in the $n$-gram can decrease performance, as is often the case with SVM machine-learning techniques (as in (Pang et al., 2002), for example).

| Corpus | # reviews | # tokens | # sentences |
|---|---|---|---|
| Thumbs Up/Down | 92 | 38,289 | 965 |
| Five Stars | 7,300 | 106,978 | 16,254 |

TAB. 3 – *Corpora Statistics.*

| Rating | % of instances |
|---|---|
| *Thumbs Down* | 67% |
| *Thumbs Up* | 33% |

TAB. 4 – *Distribution of ratings in the* Thumbs Up/Down *dataset.*

| $n$ | Features used | F-score % | | |
|---|---|---|---|---|
| | | *thumbs-down* | *thumbs-up* | overall |
| 1 | string | 85.0 | 51.7 | 78.9 |
| 1 | root | 85.1 | 50.0 | 78.9 |
| 1 | string, category | 84.2 | 50.0 | 77.8 |
| 1 | root, category | 84.1 | 50.7 | 77.8 |
| 1 | string, orth | 85.0 | 51.7 | 78.9 |
| 1 | root, orth | 85.8 | 53.0 | 80.0 |
| 1 | category, orth | 78.5 | 7.7 | 66.7 |
| 1 | string, category, orth | 84.2 | 50.0 | 77.8 |
| 1 | root, category, orth | 84.2 | 50.0 | 77.8 |
| 2 | string | 81.1 | 33.2 | 72.2 |
| 2 | root | 81.1 | 31.5 | 72.2 |
| 2 | string, orth | 81.1 | 33.2 | 72.2 |
| 2 | root, category | 80.5 | 28.2 | 71.1 |
| 2 | root, orth | 80.5 | 28.2 | 71.1 |
| 3 | string | 78.8 | 13.5 | 67.8 |
| 3 | root | 78.4 | 10.7 | 66.7 |
| 3 | root, category | 78.8 | 13.5 | 67.8 |

TAB. 5 – *Overall evaluation of* thumbs-up/down *classification using n-grams of Token features.*

| $n$ | Features used | Rating | Precision % | Recall % | F-score % |
|---|---|---|---|---|---|
| 1 | root, orth | *thumbs-down* | 77.2 | 98.8 | 85.8 |
| | | *thumbs-up* | 85.0 | 44.2 | 53.0 |
| | | overall | 80.0 | 80.0 | 80.0 |
| 1 | root | *thumbs-down* | 76.1 | 98.8 | 85.1 |
| | | *thumbs-up* | 85.0 | 40.8 | 50.0 |
| | | overall | 78.9 | 78.9 | 78.9 |
| 1 | string | *thumbs-down* | 76.2 | 98.8 | 85.0 |
| | | *thumbs-up* | 85.0 | 42.5 | 51.2 |
| | | overall | 78.9 | 78.9 | 78.9 |

TAB. 6 – *Detailed results of the best binary classifications using n-grams of token features.*

- The unigram results obtained this way compare favourably with the 74% accuracy benchmark for the binary classification of movie review texts (Turney, 2002).

Table 6 shows the detailed evaluation results by category for the three best analyses. As these breakdowns show, these experiments erred in the negative direction; i.e., it tended to miss-classify *thumbs-up* texts as *thumbs-down* more often than the other way. (This is also true for the others listed in Table 5 but not reported in more detail here.)

This directional error is understandable because the dataset is inherently biased that way (67% *thumbs-down*, as mentioned above). Nonetheless, we consider 80% overall accuracy to be a good achievement using only simple token-level features. In Table 7, we show some of the features the classifier found most valuable for identification of *thumbs-down* and *thumbs-up* texts. While one would not expect some of the identified features to be useful for classification, some of them are intuitive such as the word *no* for *thumbs-down*, and the words *excellent* and *good* for *thumbs-up*. The reader may also note that our best configuration has a tendency to produce a system with high precision but low recall for the *thumbs-up* category (the opposite tendency is observed for the *thumbs-down* category). An examination of the features considered more valuable in the trained system reveals that the features picked up are too specific and co-occurring mainly with the *thumbs-up* category resulting in high precision but with only a few instances recalled. The features picked up for the *thumbs-down* category are less specific of particular documents, therefore producing a system with higher recall for this category. The results presented here are the best we have obtained by optimising the F-score measure, depending on the application a different system which either favours precision or recall could be produced using the tools presented here. It is worth noting that we have not included stop word removal in the experiments presented here because we wanted to verify what features emerge from the data without imposing a particular bias. The SVM framework is particularly suitable for dealing with huge feature spaces as the one we are dealing with here (Joachims, 1998). Also note, that unlike other classification tasks, opinion identification may require the presence of words or expressions which in general would be considered empty of content in other contexts such as information retrieval. Here expressions such as “not”, “but”, and “didn’t” may be considered valuable for classification. In experiments carried out excluding words occurring in all documents in the corpus (e.g., maximal inverted document frequency) we have obtained a small increase in the overall F-score, but the question of what words need to be filtered out in order to produce a better word-based system needs further investigation.

| Classification | Features |
|---|---|
| *thumbs-down* | !, customer, not, that, will, the, to |
| *thumbs-up* | www, com, site, and, garage, excellent, good |

TAB. 7 – *Lexical items valuable for binary classification.*

### 6.2 Sentiment-based Classifier Results

The results of the experiment using the SVM with the more sophisticated set of features is shown in Table 8. As can be observed, the overall performance of the classifier (76%) is in absolute number lower than the lexical classifier (80%). Note however that while the lexical-based classifier is better at recognising *thumbs-down* texts, the sentiment-based classifier seems better at recognising *thumbs-up* texts. In order to verify whether the differences are significant, we run an experiment where 62 documents were used for training the algorithms and 30 documents were used for testing. The F-scores obtained at each data point (i.e., document) where compared with a *t-test*. No differences in classification accuracy were observed at a *0.05* confidence level.

In Table 9, we show some of the features the classifier found most valuable for identification of *thumbs-down* and *thumbs-up* texts using sentiment features. As can be appreciated, all features seem to play a role in classification and appear to be rather intuitive (e.g., the presence of a "negative" feature for *thumbs-down* and the absence of "negative" for *thumbs-up*).

## 7 Fine-grained Classification Experiments

The corpus of documents we are using for these experiments consisted of 600 documents containing approximately 7,300 classification instances (see Table 3), with ratings distributed unevenly as shown in Table 10. So classifying each review as 5-star would give a classification accuracy of around 68%. As in the previous experiments, two classifiers have been used: One of the classifier in a unigram based classifier which only uses lexical based features. The second set of experiments uses the more sophisticated set of features we presented before. We treated this too as a straightforward classification problem: to train the same SVM engine to assign one of the five possible features to each comment span.

### 7.1 Lexical-based Classifier Results

We carried out SVM training and evaluation with 5-fold cross-validation over the *5-star* corpus, using various combinations of token features as in the binary set of experiments.

| F-score % | | |
|---|---|---|
| *thumbs-down* | *thumbs-up* | overall |
| 82.8 | 60.6 | 76.0 |

TAB. 8 – *Overall evaluation of* thumbs-up/down *classification using a sentiment-based classifier.*

| Classification | Features |
|---|---|
| *thumbs-down* | count_neutral=8, ADV=never, count_neutral=1, senti=neg, ADJ_ADV=very late |
| *thumbs-up* | count_neg=1, count-neg=0, ADJ=good, ADJ=original, count_neutral=0, ADV=fast |

TAB. 9 – *Sentiment-based features for binary classification using sentiment-based classification.*

Because of the much greater memory and processing time required to deal with the larger corpus, and since our previous experiments had indicated (as expected) that using bigrams, trigrams, and combinations of three features would not improve the results, we limited this set of experiments to unigrams of one or two features. Table 11 summarizes the standard classification measurements for this series of experiments.

Even for five-way classification we obtained reasonably good overall results—around 74%. Unfortunately, as the detailed analysis of the two best results in Table 12 shows, the scores were very good only for the extreme classifications, *1-star* and *5-star*, whereas the scores for *2-star* and *3-star* in particular were quite low. (The detailed results for the other two experiments were similar.)

We attribute this uneven performance partly to the unbalanced distribution of ratings in our dataset (see Table 10) as well as to the inherent fuzziness of mid-range, subjective ratings. In other words, the opinions associated with *2-*, *3-*, and *4-star* ratings are less "opinionated" than *1-* and *5-star* ratings and therefore less clearly bounded. The precision and recall scores in the *2-*, *3-*, and *4-star* categories also suggest that the classification errors occur mainly within these three mid-range classes; of course, misclassifying a *3-star* text as *2-star*, for example, is much less serious than misclassifying it as *1-star*. It is also worth noting that an SVM engine treats these ratings as a set of five arbitrary strings rather than as sequential numeric values.

In Table 13, we show some of the features the classifier found most valuable for identification of the different classifications in the *5-star* corpus. It is interesting to note that extreme categories *5-star* and *1-star* are associated with very intuitive lexical items such as *excellent* and *great* for *5-stars* and *worst* and *avoid* for *1-star*.

## 7.2 Sentiment-based Classifier Results

Our second set of experiments with the *5-stars* corpus uses as before the set of features based on sentiment analysis: *syn_features*, *sentence_features*, and *text_features*. Table 14

| Rating | $\%$ of instances |
|---|---|
| *1-star* | 7.8% |
| *2-star* | 2.3% |
| *3-star* | 3.2% |
| *4-star* | 18.9% |
| *5-star* | 67.9% |

TAB. 10 – *Distribution of ratings in the* 1–5 star *dataset.*

| $n$ | Token features used | $F - score$ % by rating | | | | | |
|---|---|---|---|---|---|---|---|
| | | *1-star* | *2-star* | *3-star* | *4-star* | *5-star* | overall |
| 1 | root | 79.9 | 1.8 | 5.8 | 22.5 | 85.1 | 74.9 |
| 1 | string | 78.0 | 2.4 | 7.2 | 23.7 | 84.6 | 74.1 |
| 1 | root, category | 77.0 | 24.0 | 7.3 | 24.3 | 84.3 | 73.7 |
| 1 | root, orth | 77.8 | 4.8 | 7.6 | 23.7 | 84.8 | 74.6 |

TAB. 11 – *Overall evaluation of* 1–5 star *classification.*

| $n$ | Features used | Rating | Precision % | Recall % | $F - score$ % |
|---|---|---|---|---|---|
| 1 | root | *1-star* | 80.6 | 80.0 | 79.9 |
| | | *2-star* | 30.0 | 0.9 | 1.8 |
| | | *3-star* | 44.8 | 3.1 | 5.8 |
| | | *4-star* | 44.1 | 15.1 | 22.5 |
| | | *5-star* | 79.0 | 92.5 | 85.2 |
| | | overall | 77.0 | 73.0 | 74.9 |
| 1 | root, orth | *1-star* | 78.9 | 77.5 | 77.8 |
| | | *2-star* | 46.7 | 2.6 | 4.8 |
| | | *3-star* | 65.0 | 4.1 | 7.6 |
| | | *4-star* | 46.9 | 15.9 | 23.7 |
| | | *5-star* | 78.7 | 92.3 | 84.8 |
| | | overall | 76.6 | 72.7 | 74.6 |

TAB. 12 – *Detailed results of the best* 1–5 star *classifications.*

| Classification | Features |
|---|---|
| *1-star* | worst, not, cancelled, avoid, ... |
| *2-stars* | shirt, ball, waited, ... |
| *3-stars* | another, didnt, improve, fine, wrong, ... |
| *4-stars* | ok, test, wasnt, but, however, ... |
| *5-stars* | very, excellent, future, experience, always, great, .. |

TAB. 13 – *Lexical items valuable for fine-grained classification.*

shows results of 5-fold cross-validation experiment. The absolute performance of the classifier (72%) is lower than that of the lexical-based classifier, but one can see that the sentiment-based classifier is doing a better job in the "more difficult" 2-, 3-, and 4-star categories. Here again, we run an experiment where part of the corpus was used as training and 32 documents were used for testing. The F-scores in the testing set where compared with a *t-test*. No differences in classification accuracy were observed at a *0.05* confidence level.

| Rating | Precision % | Recall % | F-score % |
|---|---|---|---|
| *1-star* | 67.87 | 52.41 | 58.82 |
| *2-star* | 46.66 | 16.90 | 24.44 |
| *3-star* | 63.33 | 11.80 | 19.44 |
| *4-star* | 51.66 | 11.33 | 18.51 |
| *5-star* | 75.12 | 96.28 | 83.34 |
| overall | 73.72 | 71.49 | 72.58 |

TAB. 14 – *Detailed results of* 1–5 star *classifications using sentiment-based classifier.*

In Table 15, we show some of the features the classifier found most valuable for identification of the different classifications in the *5-star* corpus. Here again, the learning system seems to capture interesting and intuitive features and values: for example *5-star* is associated with the absence of "negative" words, and with positive expressions while *1-star* is associated with negation and words expressing negative sentiment. *4-stars* shows the presence of the adjective "ok" which seems rather natural while *2-stars* and *3-stars* seem more inclined towards the negative scale of sentiments. Note that all engineered features seem to play a role in classification. Finally, Table 16 shows a few examples of "vague" review texts which could explain low classification accuracy in the *2-, 3-, 4-stars* categories.

# 8 Final Remarks

Unlike many other opinion-classification studies, we decided to study first the effect of lexical information without relying on predefined word-lists or specialized lexical resources, thus allowed the machine-learning techniques used to infer the values of words implicitly from the

| Classification | Features |
|---|---|
| *1-star* | ADV_ADV=still not, ADJ=cancelled, ADJ=incorrect, ... |
| *2-stars* | count_neutral=9, ADJ=disappointing, ADV=fine, ADV=down, ADV=basically, ADV=completely, ... |
| *3-stars* | ADJ=likely, ADJ=expensive, ADJ=wrong, ADV_ADV=no able, ... |
| *4-stars* | ADV=competitive, ADJ=positive, ADJ=ok, ... |
| *5-stars* | ADV=happily, ADV=always, count_neg=0, ADV_ADJ=so simple, ADV_ADJ=very positive, ADV_ADV=not only |

TAB. 15 – *Sentiment-based features valuable for fine-grained classification.*

| Rating | Text |
|---|---|
| *2-star* | My personal details were not retained and when asked for an 'order number' on the survey I could find no trace of it. Not a very pleasing shop. I have in the past been very pleased. |
| *3-star* | Navigation is not intuitive. It seems to be logically structured but the cues are too brief or assumptive. I took twice as long as with some alternative sites. |
| *3-star* | The secure server didnt work first time so I had to go through again and reenter half my info again before it worked. It did work in the end and I hope to receive the goods soon. |

TAB. 16 – *Examples of 2- and 3-star review texts which are difficult to classify.*

training data. In a second study, we have used available lexical resources and a set of simple sentiment-based features for classification. Both approaches have acceptable performance in binary and fine-grained classification.

These approaches compare favourably to current state of the art work in the field. Our work on positive/negative distinction compares favourably to both the baseline presented in (Devitt and Ahmad, 2007) approach and their more sophisticated classifier or metrics based on lexical cohesion. In order to have a clearer picture of the situation in the fine grained classification experiments (i.e., *5-star* corpus) we have computed agreement between our classifiers and the gold standard annotation obtaining agreement of 56% for the sentiment based classifier and 65% agreement for the lexical based classifier, these results are better than agreement reported by (Devitt and Ahmad, 2007), differences might be due to the different nature of the two datasets used, ours being less complex. Our approach also compares favourably to Turney (2002) approach which obtained 74% classification accuracy. Our lexical based classifier obtained over 80% classification accuracy, note however that the differences may be due to the different characteristics of the datasets used.

We have also carried out experiments to study the effect of training using different corpus sizes: for the *thumbs-up/thumbs-down* corpus the lexical-based learner doesn't seem to be sensitive to the number of training documents: with a few documents the system achieves optimal performance; the opposite can be said of the sentiment-based classifier which improves as more documents are used for training. For the *5-star* corpus, the reverse situation has been observed: here, the lexical based classifier needs more documents to get an acceptable performance, while the sentiment-based classifier seems to be insensitive to corpus size achieving a good accuracy after seen a few documents. Further experiments need to be carried out with different training and testing partitions to verify this interesting tendency.

Finally, while this is not the focus of this paper, it is worth mentioning that our extraction technology is also being used to identify the object of the opinion as well as extracting interesting positive and negative phrases; and we are using the output of the Stanford parser and our own noun phrase chunking process in conjunction with our regular grammars in order to identify well formed phrases which contain either positive or negative words according to SentiWordNet. Table 17 shows a list of automatically identified positive and negative phrases, note that evaluation of this approach is part of our future work.

| Positive | Negative |
|---|---|
| correctly packaged bulb; their interest free sofas; just a 10% booking fee; highly recommended; a very efficient management; wonderful bargains; a good job; excellent products; a great deal; good quality; the interesting thing; a totally free service | the same disappointing experience; increasingly abusive emails; unscrupulous double glazing sales; our racist and mega incompetent police forces ; an unprofessional company ; do not buy a sofa from dfs poole or dfs anywhere; the rather poor service; a horrendous experience; excessive packaging; the worst energy supplier; not the cheapest ; such a useless company; the worse company; the bad track record; the most shockingly poor service; the utter inefficiency |

TAB. 17 – *Automatically extracted positive and negative phrases.*

Future work will include the use of more sophisticated linguistic analysis, such as dependency relations produced by the Stanford parser (Klein and Manning, 2003; de Marneffe et al., 2006). We will also experiment with segmentation of the texts and classification of segments (such as sentences), including the elimination of unopinionated segments (inspired by Pang and Lee (2004)).

# 9 Conclusions

Finding information about companies on multiple sources on the Web has become increasingly important for business analysts. In particular, since the emergence of the Web 2.0, opinions about companies and their services or products need to be found and distilled in order to create an accurate picture of a business entity and its reputation. The work presented here has been carried out in order to provide a practical solution to a business intelligence application: tracking the reputation of a company by identifying factual and subjective information about the company. We are using information extraction technology to extract company facts from multiple sources and opinion mining techniques based on supervised machine learning technology to identify positive and negative texts and fine grained sentiment classification. Although we have presented some information on factual information extraction, this paper has concentrated on the problem of opinion mining. This set of experiments indicates that we can classify short texts according to rating (the positive or negative value of the opinions) using machine-learning based on semantic and linguistic analysis. We have compared two different approaches, a lexical approach which relies on parts of speech tagging and morphological analysis, and a more sophisticated approach completely new which makes use of a lexical resource—SentiWordNet without a word sense disambiguation procedure—together with our own interpretation of the positivity and negativity scores associated to particular lexical entries. We have shown that both approaches compare very favourably to the state of the art and also have shown that although the two classifiers studied don't appear to outperform one another, interesting and intuitive features are identified by the learning algorithm. In this set of experiments, we have not concentrated on the identification of the opinion holder, because the characteristics of the dataset make that problem trivial. However, the identification of the topic of the review is not trivial and is being addressed using a syntactic and pattern-based approach. While a restricted qualitative evaluation of a prototype of the reputation teller application being described in the paper has been carried out with users and with positive outcome, full evaluation is part of our future work.

## Acknowledgements

We are indebted to three anonymous reviewers and the editors for their extensive comments, suggestions, and recommendations that helped reflect on and improve this work. This work is partially supported by the EU-funded MUSING project (IST-2004-027097).

## References

Aue, A. and M. Gamon (2005). Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*. ACL.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proc. of 3rd Conference of Applied Natural Language Processing*.

Chinchor, N. (1992). MUC-4 Evaluation Metrics. In *Proceedings of the Fourth Message Understanding Conference*, pp. 22–29.

Church, K. W. and P. Hanks (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics 16*(1), 22–29.

Cunningham, H., D. Maynard, K. Bontcheva, and V. Tablan (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).*

Dave, K., S. Lawrence, and D. M. Pennock (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, New York, NY, USA, pp. 519–528. ACM.

de Marneffe, M.-C., B. MacCartney, and C. Manning (2006). Generating typed dependency parses from phrase structure parses. In *Language Resources and Evaluation Conference*.

Devitt, A. and K. Ahmad (2007). Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 984–991. Association for Computational Linguistics.

Esuli, A. and F. Sebastiani (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, 5th Conference on Language Resources and Evaluation*, Genova, IT, pp. 417–422.

Fellbaum, C. (Ed.) (1998). *WordNet - An Electronic Lexical Database*. MIT Press.

Ghose, A., P. G. Ipeirotis, and A. Sundararajan (2007). Opinion mining using econometrics: A case study on reputation systems. In *Proceedings of the Association for Computational Linguistics*. The Association for Computational Linguistics.

Grishman, R. (1997). Information Extraction: Techniques and Challenges. In *Information Extraction: a Multidisciplinary Approach to an Emerging Information Technology*, Frascati, Italy. Springer.

Grishman, R. and B. Sundheim (1996). Message Understanding Conference - 6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen. Association for Computational Linguistics, Morristown, NJ, USA.

Hatzivassiloglou, V. and K. McKeown (1997). Predicting the semantic orientation of adjectives. In *ACL*, pp. 174–181.

Hu, M. and B. Liu (2004). Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, pp. 168–177. ACM.

Isozaki, H. and H. Kazawa (2002). Efficient Support Vector Classifiers for Named Entity Recognition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, Taipei, Taiwan, pp. 390–396.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol (Eds.), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Number 1398 in Lecture Notes in Computer Science, Chemnitz, Germany, pp. 137–142. Springer Verlag, Heidelberg.

Klein, D. and C. Manning (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Lafferty, J., A. McCallum, and F. Pereira (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco: Morgan Kaufmann, pp. 282–289.

Leek, T. R. (1997). Information Extraction Using Hidden markov Models. Technical report, University of California, San Diego, USA.

Li, Y., K. Bontcheva, and H. Cunningham (2005). SVM Based Learning System For Information Extraction. In M. N. J. Winkler and N. Lawerence (Eds.), *Deterministic and Statistical Methods in Machine Learning*, LNAI 3635, pp. 319–339. Springer Verlag.

Li, Y., K. Bontcheva, and H. Cunningham (2007a). Cost Sensitive Evaluation Measures for F-term Patent Classification. In *The First International Workshop on Evaluating Information Access (EVIA 2007)*, pp. 44–53.

Li, Y., K. Bontcheva, and H. Cunningham (2007b). SVM Based Learning System for F-term Patent Classification. In *Proceedings of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pp. 396–402.

Li, Y., K. Bontcheva, and H. Cunningham (2009). Adapting SVM for Data Sparseness and Imbalance: A Case Study on Information Extraction. *Natural Language Engineering 15*(2), 241–271.

Li, Y. and J. Shawe-Taylor (2003). The SVM with Uneven Margins and Chinese Document Categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, Singapore.

Liu, B., M. Hu, and J. Cheng (2005). Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web (WWW '05)*, New York, NY, USA, pp. 342–351. ACM.

Pang, B. and L. Lee (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics*, pp. 271–278.

Pang, B., L. Lee, and S. Vaithyanathan (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the 2002 Conference on EMNLP*, pp. 79–86.

Riloff, E., C. Schafer, and D. Yarowsky (2002). Inducing information extraction systems for new languages via cross-language projection. In *Proceedings of the 19th international conference on Computational linguistics*, Morristown, NJ, USA, pp. 1–7. Association for Computational Linguistics.

Saggion, H., A. Funk, D. Maynard, and K. Bontcheva (2007). Ontology-based information extraction for business applications. In *Proceedings of the 6th International Semantic Web Conference (ISWC 2007)*, Busan, Korea.

Tesnière, L. (1959). *Éléments de syntaxe structurale*.

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

van Rijsbergen, C. (1979). *Information Retrieval*. London: Butterworths.

Witten, I. H. and E. Frank (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

Yankova, M., H. Saggion, and H. Cunningham (2008). Adopting ontologies for multisource identity resolution. In A. Duke, M. Hepp, K. Bontcheva, and M. B. Vilain (Eds.), *OBI*, Volume 308 of *ACM International Conference Proceeding Series*, pp. 6. ACM.

Zagibalov, T. and J. Carroll (2008). Unsupervised classification of sentiment and objectivity in chinese text. In *Proceedings of IJCNLP 2008*, Hyderabad, India.

Zhuang, L., F. Jing, and X.-Y. Zhu (2006). Movie review mining and summarization. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, New York, NY, USA, pp. 43–50. ACM.

## Résumé

Dans le domaine de la veille technologique et de l'intelligence économique, la fouille de textes pour le Web joue un rôle fondamental. En particulier avec le développement du Web 2.0, les textes d'opinions sur les sociétés, leurs produits, leurs PDGs, etc. sont omniprésents sur internet. Afin de créer des profils de sociétés et de leur associer des indicateurs de reputation, les analystes économiques doivent analyser de grandes masses de données textuelles qui doivent être transformées dans des représentations structurelles avant d'efectuer un analyse plus approfondie. Dans cet article, nous utilisons des techniques d'extraction d'information pour obtenir des faits ainsi que des opinions positives et négatives associées à des sociétés. L'information est identifiée dans de multiples sources de données sur le Web et intégrée dans une base de connaissance qui est utilisée pour le raisonnement a posteriori. Cet article décrit les expériences et les résultats obtenus avec un système de classification d'opinions qui utilise des traits lexico-syntaxiques et sémantiques.