Sentiment Analysis: A Combined Approach

Rudy Prabowo[1], Mike Thelwall

School of Computing and Information Technology
University of Wolverhampton
Wulfruna Street
WV1 1SB Wolverhampton, UK
Email:rudy.prabowo@wlv.ac.uk, m.thelwall@wlv.ac.uk

Abstract

Sentiment analysis is an important current research area. This paper combines rule-based classification, supervised learning and machine learning into a new combined method. This method is tested on movie reviews, product reviews and MySpace comments. The results show that a hybrid classification can improve the classification effectiveness in terms of micro- and macro-averaged F_1. F_1 is a measure that takes both the precision and recall of a classifier's effectiveness into account. In addition, we propose a semi-automatic, complementary approach in which each classifier can contribute to other classifiers to achieve a good level of effectiveness.

Key words: sentiment analysis, unsupervised learning, machine learning, hybrid classification

1 Introduction

The sentiment found within comments, feedback or critiques provide useful indicators for many different purposes. These sentiments can be categorised either into two categories: positive and negative; or into an n-point scale, e.g., very good, good, satisfactory, bad, very bad. In this respect, a sentiment analysis task can be interpreted as a classification task where each category represents a sentiment. Sentiment analysis provides companies with a means to estimate the extent of product acceptance and to determine strategies to improve product quality. It also facilitates policy makers or politicians to analyse public sentiments with respect to policies, public services or political issues.

This paper presents the empirical results of a comparative study that evaluates the effectiveness of different classifiers, and shows that the use of multiple classifiers in a hybrid manner can improve the effectiveness of sentiment analysis. The procedure is that if one classifier fails to classify a document, the classifier will pass the document onto the next classifier, until the document is classified or no other classifier exists. Section 2 reviews a number of automatic classification techniques used in conjunction with machine learning. Section 3 lists existing work in the area of sentiment analysis. Section 4 explains the different approaches used in our comparative study. Section 5 describes the experimental method used to carry out the comparative study, and reports the results. Section 6 presents the conclusions.

2 Automatic Document Classification

In the context of automatic document classification, a set of classes, C, is required. Each class represents either a subject or a discipline.

$$C = \{c_1, c_2, c_3, ..., c_n\}$$

[1] Current address:College of Applied Sciences, Ibri, P.O. Box 14, P.C. 516, Sultanate of Oman.

where n is the number of classes in C. In addition, D is defined as a set of documents in a collection.

$$D = \{d_1, d_2, d_3, ..., d_m\}$$

where m is the number of documents in the collection. Automatic classification is defined as a process in which a classifier program determines to which class a document belongs. The main objective of a classification is to assign an appropriate class to a document with respect to a class set. The results are a set of pairs, such that each pair contains a document, d_i , and a class, c_j, where $\{d_i, c_j\} \in D \times C$. d_i, c_j means that $d_i \in D$ is assigned with (or is classified into) $c_j \in C$ (Sebastiani 2002).

In a machine learning based classification, two sets of documents are required: a training and a test set. A training set (T_r) is used by an automatic classifier to learn the differentiating characteristics of documents, and a test set (T_e) is used to validate the performance of the automatic classifier. The machine learning based classification approach focuses on optimising either a set of parameter values with respect to a set of (binary or weighted) features or a set of induced rules with respect to a set of attribute-value pairs. For example, a Support Vector Machines based approach focuses on finding a hyperplane that separates positive from negative sample sets by learning and optimising the weights of features as explained in Section 4.2. In contrast, ID3 (Quinlan 1986) and RIPPER (Cohen 1995) focus on reducing an initial large set of rules to improve the efficiency of a rule-based classifier by sacrificing a degree of effectiveness if necessary. Sebastiani (2002) states that machine learning based classification is practical since automatic classifiers can achieve a level of accuracy comparable to that achieved by human experts. On the other hand, there are some drawbacks. The approach requires a large amount of time to assign significant features and a class to each document in the training set, and to train the automatic classifier such that a set of parameter values are optimised, or a set of induced rules are correctly constructed. In the case where the number of rules required is large, the process of acquiring and defining rules can be laborious and unreliable (Dubitzky 1997). It is especially significant if we have to deal with a huge collection of web documents, and have to collect appropriate documents for a training set. There is also no guarantee that a high level of accuracy obtained in one test set can also be obtained in another test set. In this context, we empirically examine the benefits and drawbacks of machine learning based classification approaches (Section 5).

3 Existing Work in Sentiment Analysis

Whilst most researchers focus on assigning sentiments to documents, others focus on more specific tasks: finding the sentiments of words (Hatzivassiloglou & McKeown 1997), subjective expressions (Wilson et al. 2005; Kim & Hovy 2004), subjective sentences (Pang & Lee 2004) and topics (Yi et al. 2003; Nasukawa & Yi 2003; Hiroshi et al. 2004). These tasks analyse sentiment at a fine-grained level and can be used to improve the effectiveness of a sentiment classification, as shown in Pang & Lee (2004). Instead of carrying out a sentiment classification or an opinion extraction, Choi et al. (2005) focus on extracting the sources of opinions, e.g., the persons or organizations who play a crucial role in influencing other individuals' opinions. Various data sources have been used, ranging from product reviews, customer feedback, the Document Understanding Conference (DUC) corpus, the Multi-Perspective Question Answering (MPQA) corpus and the Wall Street Journal (WSJ) corpus.

To automate sentiment analysis, different approaches have been applied to predict the sentiments of words, expressions or documents. These are Natural Language Processing (NLP) and pattern-based (Yi et al. 2003; Nasukawa & Yi 2003; Hiroshi et al. 2004; König & Brill 2006), machine learning algorithms, such as Naive Bayes (NB), Maximum Entropy (ME), Support Vector Machine (SVM) (Joachims 1998), and unsupervised learning (Turney 2002).

Table 2 lists some existing work in this area, and shows different types of objectives along with the associated models used and the experimental results produced. In an ideal scenario, all the experimental results are measured based on the micro-averaged and macro-averaged precision, recall, and F_1 as explained below.

	Machine says yes	Machine says no
human says yes	tp	fn
human says no	fp	tn

Table 1: A confusion table

Precision(P) = $\frac{tp}{tp+fp}$; Recall(R) = $\frac{tp}{tp+fn}$; Accuracy(A) = $\frac{tp+tn}{tp+tn+fp+fn}$; $F_1 = \frac{2 \cdot P \cdot R}{P+R}$

Each two-by-two confusion table refers to a category that represents a sentiment. Given a set of categories, there are two different ways to measure the average performance of an automatic classifier.

1. **Micro averaging**. Given a set of confusion tables, a new two-by-two contingency table is generated. Each cell in the new table represents the sum of the number of documents from within the set of tables. Given the new table, the average performance of an automatic classifier, in terms of its precision and recall, is measured.

2. **Macro averaging**. Given a set of confusion tables, a set of values are generated. Each value represents the precision or recall of an automatic classifier for each category. Given these values, the average performance of an automatic classifier, in terms of its precision and recall, is measured.

Micro averaging treats each document equally. This means that micro averaging results in averaging over a set of documents. The performance of a classifier tends to be dominated by common classes. In contrast, macro averaging treats each class equally. This means that macro averaging results in averaging over a set of classes. The performance of a classifier tends to be dominated by infrequent classes. One class that results in a bad performance can deteriorate the overall performance significantly. Hence, it is common that macro averaged performance is lower than micro averaged performance, as shown in a classification performance evaluation conducted by Yang & Liu (1999) and Calvo & Ceccatto (2000).

The diversity of the evaluation methods and data sets used make it difficult to objectively compare the effectiveness of different approaches. Hence, we need to be cautious in interpreting the results listed in Table 1.

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

Table 2: Existing work in sentiment analysis (continued)

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

Table 1: Existing work in sentiment analysis

This paper draws upon four existing approaches.

1. NLP and pattern based approaches. This focuses on using existing natural language processing tools, such as Part-of-Speech(POS)-taggers and parsers, or N-grams, such as unigrams, bigrams and trigrams. The results generated by the tools or N-gram processors are further processed to generate a set of patterns. Each pattern is assigned a sentiment, either positive or negative. In our setting, we used the Montylingua (Liu 2004) parser to produce a collection of parsed sentences that can be further processed to form a set of rules (Section 4.1).

2. Unsupervised learning. This focuses on exploiting a search engine corpus to determine the sentiment of an expression, as demonstrated in Turney (2002). Section 4.1.3 explains our method.

3. Machine Learning. We used Support Vector Machines (SVM) (Joachims 1998), the most widely used machine learning algorithm, to measure the effectiveness of machine learning approaches. We also examined the effectiveness of two induction algorithms, ID3 (Quinlan 1986) and RIPPER (Cohen 1995).

4. Hybrid Classification The idea of hybrid classification was used in König & Brill (2006). Section 4.3 describes our hybrid classification method.

In addition, we propose a complementary approach that can be used in a real-world scenario, as illustrated in Figure 4.

4 Different Classification Approaches Used

Sections 4.1 – 4.3 explain three different classification approaches used in our comparative study. In particular, Subsections 4.1.1 – 4.1.4 describe a number of approaches that focus on acquiring and defining a set of rules (rule-based classification). Section 4.2 explains how we use Support Vector Machines in sentiment analysis. Section 4.3 explains how we use all the approaches in a hybrid manner.

4.1 Rule Based Classification

A rule consists of an antecedent and its associated consequent that have an 'if-then 'relation:

$$antecedent \Longrightarrow consequent$$

An antecedent defines a condition and consists of either a token or a sequence of tokens concatenated by the $\wedge$ operator. A token can be either a word, '?' representing a proper noun, or '#' representing a target term. A target term is a term that represents the context in which a set of documents occurs, such as the name of a politician, a policy recommendation, a company name, a brand of a product or a movie title. A consequent represents a sentiment that is either positive or negative, and is the result of meeting the condition defined by the antecedent.

$$\{token_1 \wedge token_2 \wedge \ldots \wedge token_n\} \Longrightarrow \{+|-\}$$

The two simple rules listed below depend solely on two sentiment bearing words, each of which represents an antecedent.

$$\{excellent\} \Longrightarrow \{+\}$$

$$\{absurd\} \Longrightarrow \{-\}$$

Assume that we have two sentences.

1. Laptop-A is more expensive than Laptop-B.

2. Laptop-A is more expensive than Laptop-C.

and the target word of these sentences is Laptop-A. The rule derived from these sentences is as follows:

$$\{\# \wedge \mathit{more} \wedge \mathit{expensive} \wedge \mathit{than} \wedge ?\} \Longrightarrow \{-\}$$

The interpretation of this rule is as follows: the target word, Laptop-A is less favourable than the other two laptops due to its price, which is expressed by the rule above. Here, the focus is on the price attribute of the Laptop-A.

In contrast, assume that the target words are Laptop-B and Laptop-C. The rule derived from these sentences becomes as follows:

$$\{? \wedge \mathit{more} \wedge \mathit{expensive} \wedge \mathit{than} \wedge \#\} \Longrightarrow \{+\}$$

The interpretation of this rule is as follows: the two target words, Laptop-B and Laptop-C are more favourable than the Laptop-A due to its price, which is expressed by the rule above. Here, the focus is on the price attribute of both the Laptop-B and Laptop-C.

Clearly, a target word is the crucial factor in determining the sentiment of an antecedent. In this respect, we concentrate on acquiring and defining a set of antecedents and their consequents to form a set of rules with respect to a set of target words representing the context in which a set of documents occurs, and evaluate four different classifiers, each of which applies a set of rules. We also take negation, 'not', 'neither nor'and 'no', into account. With regard to proximity, we scan all the sentences within a document, i.e., operating at sentence level. Each antecedent is then derived from a sentence.

4.1.1 General Inquirer Based Classifier (GIBC)

The first, simplest rule set was based on 3672 pre-classified words found in the General Inquirer Lexicon (Stone et al. 1966), 1598 of which were pre-classified as positive and 2074 of which were pre-classified as negative. Here, each rule depends solely on one sentiment bearing word representing an antecedent. We implemented a General Inquirer Based Classifier (GIBC) that applied the rule set to classify document collections.

4.1.2 Rule-Based Classifier (RBC)

Given a pre-classified document set, the second rule set was built by replacing each proper noun found within each sentence with '?' or '#' to form a set of antecedents, and assigning each antecedent a sentiment (the formation of a set of rules). Here, the basic assumption was that the sentiment assigned to each antecedent was equal to the sentiment assigned to the pre-classified document within which the antecedent was found. Then we implemented a Rule-Based Classifier (RBC) that applied this second rule set to classify a document collection. It is arguable that the antecedent may express a sentiment that is not the same as the associated document sentiment. Therefore, we implemented a Sentiment Analysis Tool (SAT), discussed in Section 5.4, that can be used to correct the sentiment in a semi-automatic way.

The following procedure was used to generate a set of antecedents. The Montylingua (Liu 2004) chunker was used to parse all the sentences found in the document set. Given these parsed sentences, a set of proper nouns, i.e., all terms tagged with NNP and NNPS, was automatically identified and replaced by '?'. To reduce the error rate of parsing, we automatically scanned and tested all the proper nouns identified by Montylingua against all the nouns (NN and NNS) in WordNet 2.0 (Miller 1995). When WordNet regarded the proper nouns as standard nouns, the proper nouns were regarded as incorrectly tagged, and were not replaced with '?'. In addition, all target words were replaced with '#'. As a result, a set of antecedents was generated. A suffix array (Manber & Myers 1990) was then built to speed up antecedent matching.

4.1.3 Statistics Based Classifier (SBC)

The Statistics Based Classifier (SBC) used a rule set built using the following assumption. Bad expressions co-occur more frequently with the word 'poor', and good expressions with the word 'excellent'(Turney 2002). We calculated the closeness between an antecedent representing an expression and a set of sentiment bearing words. The following procedure was used to statistically determine the consequent of an antecedent.

1. Select 120 positive words, such as amazing, awesome, beautiful, and 120 negative words, such as absurd, angry, anguish, from the General Inquirer Lexicon.

2. Compose 240 search engine queries per antecedent; each query combines an antecedent and a sentiment bearing word.

3. Collect the hit counts of all queries by using the Google and Yahoo search engines. Two search engines were used to determine whether the hit counts were influenced by the coverage and accuracy level of a single search engine. For each query, we expected the search engines to return the hit count of a number of Web pages that contains both the antecedent and a sentiment bearing word. In this regard, the proximity of the antecedent and word is at the page level. A better level of precision may be obtained if the proximity checking can be carried out at the sentence level. This would lead to an ethical issue, however, because we have to download each page from the search engines and store it locally for further analysis.

4. Collect the hit counts of each sentiment-bearing word and each antecedent.

5. Use 4 closeness measures to measure the closeness between each antecedent and 120 positive words (S^+) and between each antecedent and 120 negative words (S^-) based on all the hit counts collected.

$$S^+ = \sum_{i=1}^{120} Closeness(antecedent, word_i^+) \tag{1}$$

$$S^- = \sum_{i=1}^{120} Closeness(antecedent, word_i^-) \tag{2}$$

If the antecedent co-occurs more frequently with the 120 positive words ($S^+ > S^-$), then this would mean that the antecedent has a positive consequent. If it is vice versa ($S^+ < S^-$), then the antecedent has a negative consequent. Otherwise ($S^+ == S^-$), the antecedent has a neutral consequent. The closeness measures used are described below.

Document Frequency (*DF*). This counts the number of Web pages containing a pair of an antecedent and a sentiment bearing word, i.e., the hit count returned by a search engine. The larger a *DF* value, the greater the association strength between *antecedent* and *word*. The use of *DF* in the context of automatic classification can be found in Yang & Pedersen (1997) and Yang (1999).

The other three measures can be formulated based on the 2x2 contingency table shown below.

	antecedent	$\overline{antecedent}$	
word	a	b	r_1 = a + b
$\overline{word}$	c	d	r_2 = c + d
	c_1 = a + c	c_2 = b + d	N = a + b + c + d

Table 2: A contingency table for counts of co-occurrences of a sentiment-bearing word and an antecedent within a set of *N* documents.

Mutual Information (*MI*). The *MI* value of an antecedent with respect to a sentiment bearing word is computed as follows:

$$MI = log_2 \frac{P(word, antecedent)}{P(word) \cdot P(antecedent)} = log_2 \frac{a \cdot N}{(a+b) \cdot (a+c)} \tag{3}$$

The larger an *MI* value, the greater the association strength between *antecedent* and *word*, where $MI(antecedent, word)$ must be > 0. This means that the joint probability, $P(antecedent, word)$ must be greater than the product of the probability of $P(antecedent)$ and $P(word)$. Two examples of the use of this method for measuring the strength of two terms association can be found in Conrad & Utt (1994) and Church & Hanks (1989).

Chi-square (χ^2). Given two 2x2 contingency tables, with one table containing observed frequencies and another containing expected frequencies, the χ^2 value of a word with respect to an antecedent is computed as follows:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{4}$$

where $i = \{a \ldots d\}$ represents the value of each cell in a 2x2 contingency table. The Yates continuity correction is applied to each χ^2 calculation as the degree of freedom is 1. The χ^2 calculation used in this experiment does not approximate the χ^2 value, such as described in Yang & Pedersen (1997) and Swan & Allan (2000). The larger a χ^2 value, the stronger the evidence to reject the null hypothesis, which means that *word* and *antecedent* are dependent on each other. For the χ^2-test, in order to reliably accept or reject H_0, the expected values should be > 5. Otherwise, it tends to underestimate small probabilities, which incorrectly results in accepting H_1 (Cochran 1954).

Log Likelihood Ratio ($-2 \cdot log\lambda$). The log-likelihood ratio is computed as follows:

$$\begin{aligned} -2 \cdot log\lambda &= 2 \cdot \left\{ \sum_i O_i \cdot ln \frac{O_i}{E_i} \right\} \\ -2 \cdot log\lambda &= 2 \cdot \left\{ \sum_i i \cdot ln(i) + N \cdot lnN - \sum_j j \cdot ln(j) \right\} \end{aligned} \tag{5}$$

$i = \{a, b, c, d\}$ and $j = \{c_1, c_2, r_1, r_2\}$. Log likelihood ratio (Dunning 1993) follows the χ^2 hypothesis, i.e., the larger a log likelihood ratio value, the stronger the evidence to reject the null hypothesis, which means that *word* and *antecedent* are dependent on each other. Unlike χ^2, the log likelihood ratio is more accurate than χ^2 for handling rare events. As a ranking function, the log likelihood ratio is therefore a better measure than χ^2 for handling rare events.

The following gives an example of the use of the closeness measures explained above. Assume that we have the data listed in Table 3.

	antecedent	$\overline{antecedent}$	
word	40	10	$r_1 = 50$
$\overline{word}$	10	40	$r_2 = 50$
	$c_1 = 50$	$c_2 = 50$	$N = 100$

Table 3: An example of a contingency table for counts of co-occurrences of a sentiment-bearing word and an antecedent within a set of *N* documents.

- $DF = 40$, i.e., a = the number of documents in which both the antecedent and word occur.
- $MI = log_2 \frac{a \cdot N}{(a+b) \cdot (a+c)} = log_2 \frac{40 \cdot 100}{50 \cdot 50} = 0.68$.

- Prior to calculating the χ^2, the expected frequencies, E are calculated.

	antecedent	$\overline{antecedent}$
word	25	25
$\overline{word}$	25	25

Table 4: The expected frequencies, E.

$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} = 36.$

- Log likelihood ratio = $2 \cdot \left\{\sum_i i \cdot ln(i) + N \cdot lnN - \sum_j j \cdot ln(j)\right\} = 38.55.$

Both χ^2 and log likelihood ratio strongly indicate that there is a sufficient evidence to reject H_0. This means that there is a degree of dependence between the antecedent and word.

4.1.4 Induction Rule Based Classifier (IRBC)

Given the two rule sets generated by the Rule Based Classifier (RBC) and Statistics Based Classifier (SBC), we applied two existing induction algorithms, ID3 (Quinlan 1986) and RIPPER (Cohen 1995) provided by Weka (Witten & Frank 2005), to generate two induced rule sets, and built a classifier that could use the two induced rule sets to classify a document collection.

These two induced rule sets can hint about how well an induction algorithm works on an uncontrolled antecedent set, in the sense that the antecedent tokens representing attributes are not predefined, but simply derived from a pre-classified document set. The expected result of using an induction algorithm was to have both an efficient rule set and better effectiveness in terms of both precision and recall.

4.2 Support Vector Machines

We used Support Vector Machine (SVM^{light}) V6.01 (Joachims 1998). As explained in Dumais & Chen (2000) and Pang et al. (2002) given a category set, $C = \{+1, -1\}$ and two pre-classified training sets, i.e., a positive sample set, $T_r^+ = \sum_{i=1}^{n}(d_i, +1)$ and a negative sample set, $T_r^- = \sum_{i=1}^{n}(d_i, -1)$, the SVM finds a hyperplane that separates the two sets with maximum margin (or the largest possible distance from both sets), as illustrated in Figure 1. At pre-processing step, each training sample is converted into a real vector, x_i that consists of a set of significant features representing the associated document, d_i. Hence, $Tr^+ = \sum_{i=1}^{n}(x_i, +1)$ for the positive sample set and $Tr^- = \sum_{i=1}^{n}(x_i, -1)$ for the negative sample set.

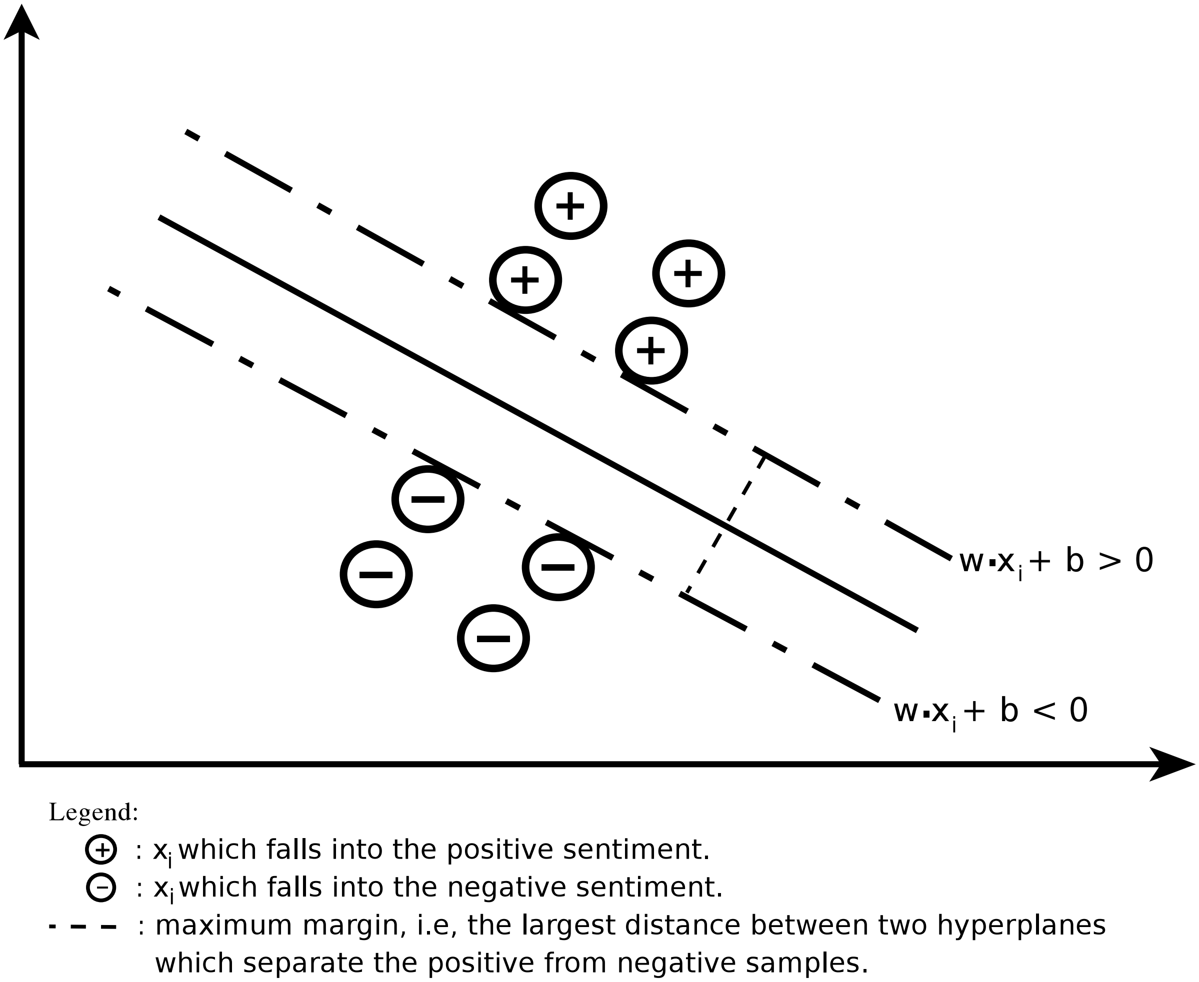

Figure 1: An illustration of the SVM method.

In this regard, for c_i=+1, $w \cdot x_i + b > 0$, and for c_i=-1, $w \cdot x_i + b < 0$. Hence, $\forall_{T_r^+, T_r^-}\{c_i \cdot (w \cdot x_i + b) \geq 1\}$ This becomes an optimisation problem defined as follows: minimise $\frac{1}{2} \cdot \|w\|^2$, subject to $c_i \cdot (w \cdot x_i + b) \geq 1$. The result is a hyperplane that has the largest distance to x_i from both sides. The classification task can then be formulated as discovering which side of the hyperplane a test sample falls into. In an ideal scenario, we expect to find a clear separation between a positive and a negative set, in the sense that the significant features found within a positive set do not appear in a negative set, or all the features position do not cross over their associated hyperplane. In real-world scenarios, this clear-cut scenario is highly unlikely, however. One document may well contain features that appear in both sets. This would mean that the features can fall into the wrong side. To handle this issue, the SVM allows the features to be included, but penalises them to indicate that x_i contains some features that fall into the wrong side and these features should not dominate the classification decision. For extremely noisy data, when there are a number of features which strongly indicate that the associated document belongs to both positive and negative sentiments, the SVM may fail. To illustrate this issue, assume that we have two positive and two negative training samples, each of which contains a number of features. Here, we categorise ft_3 into both categories, i.e., introducing noise into the training set.

$tr_1^+ = \{ft_1, ft_2\}$.
$tr_2^+ = \{ft_1, ft_3\}$.
$tr_3^- = \{ft_3, ft_4\}$.
$tr_4^- = \{ft_4, ft_5\}$.

Now, assume that we have two test samples that should be categorised as positive and one test sample as negative. Each test sample along with its associated features is listed below. Here, te^- means that the test sample should be categorised as negative, and te^+ indicates positive.

$te_1^+ = \{ft_1, ft_2, ft_4\}$. The sample contains two features that refer to positive sentiment, and only one feature for negative sentiment (relatively noisy data).

$te_2^+ = \{ft_1, ft_2, ft_4, ft_5\}$. The sample contains features that refer to both positive and negative sentiment (a very noisy data).

$te_3^- = \{ft_3\}$. The sample only contains one feature that can refer to either positive or negative sentiment (sparse and noisy data).

The classification results are as follows: for te_1^+, the SVM can correctly determine the side of the hyperplane the test sample falls into, although it contains ft_4. For the te_2^+ and te_3^-, the SVM fails to correctly classify the samples due to a high level of ambiguity and sparseness in the test samples.

In our setting, given a pre-classified document set, we automatically converted all the characters into lower cases, and carried out tokenisation. Given all the tokens found, a set of significant features was selected by using a feature selection method, i.e., document frequency as used by Pang et al. (2002), so that we can compare our results with some existing results. As observed by Dumais & Chen (2000) and Pang et al. (2002), to improve the performance of the SVM, the frequencies of all the features within each document should be treated as binary and then normalised (document-length normalisation).

4.3 Hybrid Classification

Hybrid classification means applying classifiers in sequence. A set of possible hybrid classification configurations is listed below.

1. $RBC \rightarrow GIBC$
2. $RBC \rightarrow SBC$
3. $RBC \rightarrow SVM$
4. $RBC \rightarrow GIBC \rightarrow SVM$
5. $RBC \rightarrow SBC \rightarrow GIBC$
6. $RBC \rightarrow SBC \rightarrow SVM$
7. $RBC \longrightarrow SBC \rightarrow GIBC \rightarrow SVM$
8. $RBC_{induced} \rightarrow SBC \rightarrow GIBC \rightarrow SVM$
9. $RBC \rightarrow SBC_{induced} \rightarrow GIBC \rightarrow SVM$
10. $RBC_{induced} \rightarrow SBC_{induced} \rightarrow GIBC \rightarrow SVM$

The rule set used by the RBC was directly derived from a pre-classified document set, and had a high level of precision when it was applied to a test set. Hence, it was placed first. The antecedent set used by the SBC was also directly derived from a pre-classified document set. Hence, the antecedent set had a high level of expressiveness, and was much better than the antecedent set used by the GIBC, which was quite sparse. These are the reasons for the 2^{nd} configuration. The SVM classifier was placed last because all the documents classified by the SVM were classified into either positive or negative. Hence, it did not give another classifier the chance to carry out a classification once applied. The 7^{th} configuration is the longest sequence and applies all the existing classifiers. Based on the 7^{th} configuration, the $8^{th} - 10^{th}$ configurations were defined with respect to the two induced rule sets, discussed in Subsection 4.1.4, to evaluate the effectiveness of each induction algorithm used with respect to each induced rule set.

5 Experiment

This section describes the experiment and lists the experimental results.

5.1 Data

To evaluate the effectiveness of all the approaches used, we collected three sets of samples listed below.

Population	Samples	# of Features	# of RBC Rules	# of SBC Rules
1. Movie Reviews	S1: 1000(+) & 1000(–)	44140	37356	35462
2. Movie Reviews	S2: 100(+) & 100(–)	14627	3522	2033
3. Product Reviews	S3: 180(+) & 180(–)	2628	969	439
4. MySpace Comments	S4: 110(+) & 110(–)	1112	384	373

Table 5: The data sets

The second column refers to the number of pre-classified samples: 50% of which were classified into positive and 50% as negative. The third column refers to the number of features extracted from the sample set (Section 4.2). The fourth column refers to the number of rules used by the RBC classifier (Section 4.1.2). This rule set was derived from a training set. The fifth column refers to the number of rules used by the SBC classifier (Section 4.1.3). This rule set was derived from a test set that could not be classified by the RBC.

The first data set was downloaded from Pang (2007). The second data set was a small version of the first data set, i.e., the first 200 samples extracted from the first data set. This small data set was required to test the effectiveness of the SBC and the induction algorithms (Section 4.1.4), which could not handle a large data set. The third data set was proprietary data provided by Market-Sentinel (2007). This is a clean data set that was pre-classified by experts. The fourth data set was also proprietary, provided by Thelwall (2008), extracted from MySpace (2007), and preclassified by three assessors with kappa(κ) = 100%, i.e. the three assessors completely agreed with each other. Whilst the movie review data contains a lot of sentences per document, the product reviews and MySpace comments are quite sparse.

5.2 Experimental Procedure

Figure 2 illustrates the experimental procedure. For each sample set, we carried out 10-fold cross validation. For each fold, the associated samples were divided into a training and a test set. For each test sample, we carried out a hybrid classification, i.e., if one classifier fails to classify a document, the classifier passes the document onto the next classifier, until the document is classified or no other classifier exists. Given a training set, the RBC used a Rule Generator to generate a set of rules and a set of antecedents to represent the test sample and used the rule set derived from the training set to classify the test sample. If the test sample was unclassified, the RBC passed the associated antecedents onto the SBC, which used the Closeness Measurer to determine the consequents of the antecedents. Prior to applying a closeness measure for each antecedent, the Closeness Measurer submitted a set of queries and collected the associated hit counts. If the SBC could not classify the test sample, the SBC passed the associated antecedents onto the GIBC, which used the 3672 simple rules to determine the consequents of the antecedents. The SVM was given a training set to classify the test sample if the three classifiers failed to classify it. The classification result was sent to and stored by the 10-fold cross validator to produce an evaluation result in terms of micro-averaged F_1 and macro-averaged F_1.

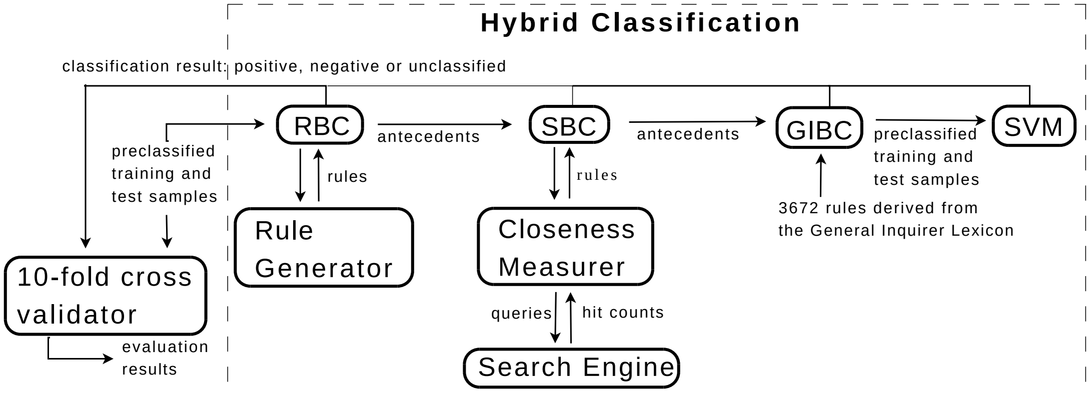

Figure 2: The experimental procedure.

Figure 3 shows the added and modified parts used to test the effect of an induction algorithm on the effectiveness of rule based classification

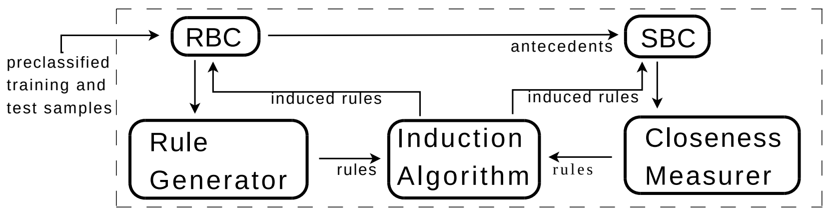

Figure 3: The added and modified parts of the experimental procedure.

5.3 Experimental Results

Table 6 lists the micro-averaged F_1 and macro-averaged F_1 of all the hybrid classification configurations and of SVM only. The results indicate that the SBC improved the effectiveness of hybrid classification, especially the hybrid configuration in which the SBC and SVM were applied. The SBC was not applied for the sample set 1 to (1) avoid ethical issues, i.e., hitting search engines with a significant amount of queries (= 8,510,880 queries), and (2) save a significant amount of time for the Closeness Measurer to collect the hit counts required.

The results also indicate that the GIBC could not always improve the effectiveness of hybrid classification. The GIBC could even reduce the effectiveness of hybrid classification.

The third sample set did not have sufficient positive and negative exemplars in the training set. This made the SVM too weak to correctly classify test samples. There was also a reduction in SVM effectiveness for the second sample set, which was the small version of the first data set. These indicate that SVM requires a significant amount of positive and negative exemplars, which should appear in both the training and test sets to achieve a high level of effectiveness.

Micro-averaged F_1 ; Macro-averaged F_1								
	RBC→GIBC	RBC→SBC	RBC→SVM	RBC→GIBC →SVM	RBC→SBC →GIBC	RBC→SBC →SVM	RBC→SBC →GIBC→SVM	SVM
S1	58.80; 58.35	n/a	83.35; 83.33	66.75; 66.40	n/a	n/a	n/a	**87.30; 87.29**
S2	57.00; 56.04	84.50; 84.17	76.00; 75.83	67.00; 66.41	88.00; 87.75	**91.00; 90.78**	89.00; 88.77	75.50; 75.32
S3	46.39; 45.30	67.22; 66.67	57.78; 56.87	63.33; 62.82	78.33; 78.22	82.78; 82.57	**83.33; 83.26**	56.94; 55.90
S4	58.64; 56.62	83.18; 82.96	84.09; 83.86	78.18; 77.65	89.09; 89.02	90.00; 89.89	**90.45; 90.38**	84.09; 83.86

Table 6: The hybrid classification evaluation results

Table 7 lists the effectiveness of each closeness measure used by the SBC to determine the consequent of each antecedent with respect to the two search engines used: Yahoo! and Google. Surprisingly, the results clearly show that the search engine used did affect each closeness measure. In addition, the use of Google resulted in a better effectiveness than that of Yahoo! for the third and fourth data sets, and the log likelihood ratio performed better than the other three closeness measures for all the sample sets used. For the second sample set, Yahoo! performed better than Google, however. In our experimental setting, the SBC used Google hit counts and the log likelihood ratio.

	Micro-averaged F_1; Macro-averaged F_1							
	Google				Yahoo			
	DF	MI	χ^2	LLR	DF	MI	χ^2	LLR
2033 rules of S2	67.29; 67.11	57.85; 54.99	56.47; 54.81	75.41; 75.20	65.42; 65.36	67.09; 63.62	63.65; 62.23	**83.33; 82.29**
439 rules of S3	73.02; 71.16	64.40; 64.36	64.85; 64.83	**79.59; 79.22**	74.60; 73.37	66.89; 66.89	63.04; 63.01	78.68; 78.42
373 rules of S4	64.63; 62.53	60.90; 60.34	64.36; 64.13	**71.04; 70.32**	64.63; 62.53	60.37; 59.85	63.83; 63.63	70.41; 69.79

Table 7: The evaluation results of the closeness measures

Table 8 lists the results of hybrid classifications that used all the classifiers available, with respect to the two induction algorithms used as illustrated in Figure 3. The results clearly show a reduction in the effectiveness of hybrid classification due to the induced rules used. The best results were produced by using ID3 and the SBC rule set. ID3 was less aggressive than RIPPER, as shown in Table 9, which lists the number of rules and attributes before and after induction, along with the time complexity of the algorithms. Hence, ID3 was more effective than RIPPER in terms of micro- and macro-averaged F_1. ID3 needed much more time than RIPPER to generate an induced rule set, however. In our setting, we ran RIPPER twice. For the first run, we used the default parameters. For the second run, we deactivated pruning. The results show no clear effect of pruning on RIPPER's effectiveness.

Micro-averaged F_1; Macro-averaged F_1			
Algorithm	$RBC_{induced}$ →SBC →GIBC→SVM	RBC→ $SBC_{induced}$ →GIBC→SVM	$RBC_{induced}$ → $SBC_{induced}$ →GIBC→SVM
S2			
ID3	50.50; 34.36	**85.50; 85.27**	50.50; 34.36
Ripper	50.00; 35.40	51.50; 37.77	49.50; 34.96
Ripper(no pruning)	51.50; 36.86	51.50; 37.77	51.50; 36.86
S3			
ID3	54.44; 41.20	**72.22; 70.83**	53.61; 40.52
Ripper	53.61; 38.61	53.33; 41.18	53.33; 38.33
Ripper(no pruning)	50.28; 33.92	55.28; 46.77	50.28; 33.92
S4			
ID3	52.27; 37.68	**90.00; 89.92**	52.27; 37.68
Ripper	50.00; 33.33	67.27; 62.80	50.00; 33.33
Ripper(no pruning)	50.00; 33.33	67.27; 62.80	50.00; 33.33

Table 8: The evaluation results with respect to the two induction algorithms used

	RBC		SBC	
	# of induced rules & attributes	time	# of induced rules & attributes	time
S2	3522 rules; 6799 attributes		2033 rules; 4404 attributes	
ID3	1701 rules; 1698 attributes	1.67 hours	1349 rules; 1326 attributes	21.16 mins
Ripper	4 rules; 3 attributes	99.9 secs	4 rules; 3 attributes	20.13 secs
Ripper(no pruning)	11 rules; 15 attributes	1.32 mins	9 rules; 21 attributes	18.66 secs
S3	969 rules; 1687 attributes		439 rules; 1010 attributes	
ID3	377 rules; 372 attributes	30 secs	190 rules; 189 attributes	14.74 secs
Ripper	1 rules; 0 attributes	3.62 secs	4 rules; 3 attributes	2.1 secs
Ripper(no pruning)	4 rules; 7 attributes	3 secs	10 rules; 21 attributes	1.7 secs
S4	384 rules; 435 attributes		373 rules; 622 attributes	
ID3	163 rules; 162 attributes	6.9 secs	153 rules; 152 attributes	8.2 secs
Ripper	1 rules; 0 attributes	2.3 secs	4 rules; 3 attributes	1.24 secs
Ripper(no pruning)	2 rules; 1 attributes	1.2 secs	8 rules; 26 attributes	0.89 secs

Table 9: The number of the induced RBC and SBC rules and attributes and the time complexity of the algorithms used

5.4 Discussion

The rule based classifiers (Section 4.1.1–4.1.4) are fundamentally different from the SVM classifier (Section 4.2) in terms of their underlying mechanics. A rule-based approach regards a collection of documents as a mining field from which a set of antecedents (or patterns) can be extracted, optimised and stored in an efficient way for pattern-query matching. The patterns are extracted by using either a set of predefined templates or heuristic methods.We simply replaced each proper noun found within each sentence with '?' or '#' to form a pattern set (Section 4.1.2). The mappings between each pattern and a category lead to the construction and optimisation of a rule set. Therefore, given a training set, the associated model focuses on either extending the existing rule set to cope with an unseen sample set (Section 4.1.3) or optimising the existing rules to enable an efficient matching (rule induction) (Section 4.1.4). A rule-based classifier simply makes use of the model to find the mappings between a pattern set and the associated categories.

In contrast, a SVM classifier, like other parametric approaches, such as Naive Bayes (Gövert et al. 1999), Rocchio (Ittner et al. 1995), and Neural network (Yin & Savio 1996) classifiers, regards a document collection as a set of significant features, each of which is assigned a weight, and each document is represented by a feature set. Therefore, given a training set, the associated model focuses on optimising the weights and other parameters required, such that the model can achieve a high level of effectiveness on an unseen sample set. Here, the independence of features is assumed, which is not always true, as explained in Lewis (1998) and Belew (2000). In addition, we, sometimes, have to make a trade-off between the effectiveness of the model and the time needed to train the model. An efficient algorithm is usually required to train the model. The associated classifier simply makes use of the model containing the learned weights and parameters to classify an unseen sample set.

In the context of real-world scenarios, the results listed in Table 6 suggest the following. If dealing with a web document collection with sufficient human classifiers to both produce a large scale of training set and make sure that a set of sufficient positive and negative exemplars appears in both the training set and the associated test set, SVM is the best choice to get good results, because SVM can efficiently and effectively process the collection.A problem arises if we want to know the expressions, which are represented by antecedents, as the reason for selecting a category in order to carry out a further deeper analysis, because SVM operates at the feature level but not at the antecedent level. To deal with a dynamic collection, such as a Web collection, the training set needs to be updated. The cost of hiring human classifiers, who may need a considerable amount of time to extend/update the existing training set, and the time required to train the model could be considerable.

In contrast, if we cannot hire human classifiers to produce a training set, we can apply the SBC and a closeness measure to use a corpus to determine the consequents of antecedents. This has three advantages. First, no human classifiers are required to produce a training set. Hence, it significantly reduces cost and time. Second, the SBC can show the antecedents as the reason for selecting an appropriate category, and assign the associated antecedent to the category. Third, by using a human classifier to judge the assigned consequent, we can build a rule database that grows over time. To build this rule database, we created a Sentiment Analysis Tool (SAT) that can assist a human classifier in checking the correctness of a new rule and integrating it into the existing rule database if it does not exist. By using this rule database, the SAT can also build a training set, which can be exploited by the SVM to classify the documents that cannot be classified by both the RBC and SBC. In this respect, the use of RBC, SBC, and SVM in a hybrid and semi-automatic manner can be interpreted as a complementary approach, i.e., each classifier contributes to other classifiers to achieve a good level of effectiveness, as illustrated in Figure 4. The SBC and SAT provide the RBC with a rule database, and the SVM with a training set. As a result, the RBC and SVM can assist the SBC to achieve a better level of effectiveness and efficiency. The problem arises if we do not have our own relatively large corpus, because we will overload a search engine with a huge amount of queries (an ethical issue) and spend a lot of time to collect the hit counts required (an efficiency issue). Another problem is to deal with both the coverage level and fluctuation of a search engine, which can affect the effectiveness of the SBC as shown in Table 7. The coverage issues are discussed in Bar-Ilan (2001) and Thelwall (2000), and the fluctuation issue in Bar-Ilan (1999) and Mettrop & Nieuwenhuysen (2001). The fluctuation and ethical issues are the main reasons for not being able to collect the hit counts for the S1 sample set.

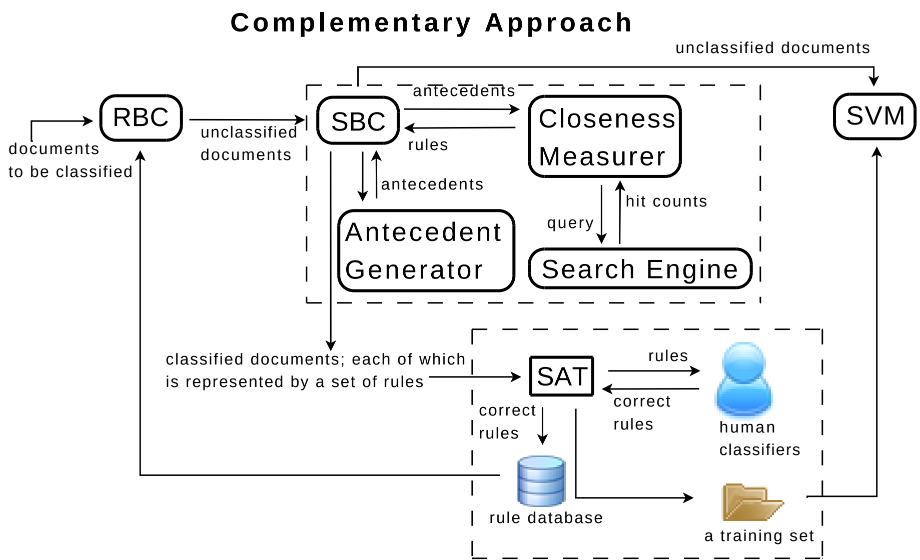

Figure 4: A diagram illustrating a complementary approach.

This means that no classifier outperforms other classifiers: they need each other to achieve the best performance.

In our experimental setting, we decided to assign a document one sentiment only (binary classification), so that F_1 measure could be applied without the risk of over-fitting. The rule-based classifiers used can carry out a multiple classification, however, i.e., assigning a document more than one category. The classifiers classify each sentence within a document, and then rank all possible sentiments in descending order. For a binary classification, the classifiers only select the top rank. For a multiple classification, the classifiers can select the best n categories, where $n \geq 1$. A further extension of the system would be to use it to create fuzzy classifications (Kuncheva 2000), i.e., assigning sentiment on a probability rather than a binary basis. Fuzzy hybrid classification has been achieved before (Ishibuchi et al. 2000) and so this should be possible for our system.

In regard to proximity, at the current stage, we scan all the sentences within a document (sentence level): each antecedent is then derived from a sentence. Although operating at the abstract or keyword level can improve efficiency, our preliminary observation revealed that we often missed the parts that could lead us to a correct category. Another possibility that we would like to examine in the near future is to crystallise the process of sentiment categorization by analysing each paragraph/section, or by having a modular domain ontology, which can allow a classifier to group relevant information together prior to assigning an appropriate sentiment. This will require a robust segmentation techniques and a concept extraction algorithm that can make use of a set of domain ontologies.

In regard to the sample sets used, as explained in Section 5.1, two types of sample set were used: one that is a collection of long documents, and another of much shorter documents. The hybrid classification performed best for S2–S4. In our setting, we selected corpuses that may well contain sentiment expressions or sentiment-bearing words because our aim is to evaluate different classification approaches used for sentiment analysis. Despite this limitation, we would argue that the combination of rule-based classifiers and a SVM classifier in a hybrid manner is likely to also perform best for other types of samples because the defining factor is not the sentiment expressions or words, but a set of well-defined patterns and features that can lead to the construction of an optimal and efficient classification model.

The use of the RIPPER algorithm resulted in a significant decrease in terms of micro- and macro-averaged F_1 as shown in Table 8 due to its high level of aggression as shown in Table 9. In contrast, the use of the ID3 algorithm decreased the effectiveness of the hybrid classification significantly less than RIPPER, i.e., between 0.45 and 11.11 in terms of micro-averaged F_1 and between 0.46 and 12.43 in terms of macro-averaged F_1. The proportion of the ID3 reduction in terms of the number of reduced rules was between 33.64% and 58.98%. Although this significant reduction only resulted in a slight decrease in effectiveness, the induction algorithm generated a set of induced antecedents that are too sparse for a deeper analysis. In a real-world scenario, it is desirable to have two rule sets, one is the original set, and another one is the induced rule set.

6 Conclusions

The use of multiple classifiers in a hybrid manner can result in a better effectiveness in terms of micro- and macro-averaged F_1 than any individual classifier. By using a Sentiment Analysis Tool (SAT), we can apply a semi-automatic, complementary approach, i.e., each classifier contributes to other classifiers to achieve a good level of effectiveness. Moreover, a high level of reduction in terms of the number of induced rules can result in a low level of effectiveness in terms of micro- and macro-averaged F_1. The induction algorithm can generate a set of induced antecedents that are too sparse for a deeper analysis. Therefore, in a real-world scenario, it is desirable to have two rule sets, one is the original set, and another one is the induced rule set.

Acknowledgements. The work was supported by a European Union grant for activity code NEST-2003-Path-1 and the Future & Emerging Technologies scheme. It is part of the CREEN (Critical Events in Evolving Networks, contract 012684) and CyberEmotions projects. We would like to thank Mark Rogers of Market Sentinel for help with providing classified data.

References

Bar-Ilan, J. (1999). Search engine results over time: A case study on search engine stability. *Cybermetrics*, 2/3(1).

Bar-Ilan, J. (2001). Data collection methods on the Web for informetric purposes: A review and analysis. *Scientometrics*, 50(1), 7–32.

Belew, R. K. (2000). *Finding out about - a cognitive perspective on search engine technology and the WWW*. Cambridge University Press, 1st edition.

Calvo, R. A. & Ceccatto, H. A. (2000). Intelligent document classification. *Intelligent Data Analysis*, 4(5), 411–420.

Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceeding of the conference on empirical methods in natural language processing (EMNLP 2005), October 6–8, 2005* (pp. 355–362). Vancouver, B.C., Canada.

Church, K. W. & Hanks, P. (1989). Word association norms, mutual information and lexicography. In *Proceedings of the 27th annual meeting of the Association for Computational Linguistics (ACL), June 26–29, 1989* (pp. 76–83). Vancouver, B.C.

Cochran, W. G. (1954). Some methods for strengthening the common χ^2 tests. *Biometrics*, 10, 417–451.

Cohen, W. W. (1995). Fast effective rule induction. In A. Prieditis & S. Russell (Eds.), *Proceedings of the 12th international conference on machine learning (ICML 1995), July 9–12, 1995* (pp. 115–123). Tahoe City, California, USA.

Conrad, J. G. & Utt, M. H. (1994). A sytem for discovering relationships by feature extraction from Text Databases. In W. B. Croft & C. J. van Rijsbergen (Eds.), *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval, July 3–6, 1994* (pp. 260–270). Dublin, Ireland.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international WWW conference, May 20–24, 2003* (pp. 519–528). Budapest, Hungary.

Dubitzky, W. (1997). *Knowledge integration in case-based reasoning: a concept-centred approach*. PhD thesis, University of Ulster.

Dumais, S. & Chen, H. (2000). Hierarchical classification of Web content. In E. Yannakoudakis, N. J. Belkin, M.-K. Leong, & P. Ingwersen (Eds.), *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, July 24–28, 2000* (pp. 256–263). Athens, Greece.

Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 61–74.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on computational linguistics (COLING 2004), August 23 – 27, 2004* (pp. 841–847). Geneva, Switzerland.

Gövert, N., Lalmas, M., & Fuhr, N. (1999). A probabilistic description-oriented approach for categorising Web documents. In S. Gauch & I.-Y. Soong (Eds.), *Proceedings of the 8th international conference on information and knowledge management (CIKM 1999), November, 1999* (pp. 474–482). Kansas City, USA.

Hatzivassiloglou, V. & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 8th conference on european chapter of the association for computational linguistics* (pp. 174–181). Madrid, Spain.

Hiroshi, K., Tetsuya, N., & Hideo, W. (2004). Deeper sentiment analysis using machine translation technology. In *Proceedings of the 20th international conference on computational linguistics (COLING 2004), August 23 – 27, 2004* (pp. 494–500). Geneva, Switzerland.

Ishibuchi, H., Nakashima, T., & Kuroda, T. (2000). A hybrid fuzzy gbml algorithm for designing compact fuzzy rule-based classification systems. *The 9th IEEE International Conference on Fuzzy Systems*, 2(706–711).

Ittner, D. J., Lewis, D. D., & Ahn, D. D. (1995). Text categorization of low quality images. In *Proceedings of the 4th annual symposium on document analysis and information retrieval (SDAIR 1995), April, 1995* (pp. 301–315). Las Vegas, USA.

Joachims, T. (1998). Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods: support vector learning*. The MIT Press.

Kim, S.-M. & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20^{th} international conference on computational linguistics (COLING 2004), August 23 – 27, 2004* (pp. 1367–1373). Geneva, Switzerland.

König, A. C. & Brill, E. (2006). Reducing the human overhead in text categorization. In *Proceedings of the 12^{th} ACM SIGKDD conference on knowledge discovery and data mining, August 20–23, 2006* (pp. 598–603). Philadelphia, Pennsylvania,USA.

Kuncheva, L. I. (2000). *Fuzzy Classifier Design*. springer, 1^{st} edition.

Lewis, D. D. (1998). Naive Bayes at forty: the independence assumption in information retrieval. In C. Nédellec & C. Rouveirol (Eds.), *Proceedings of the 10^{th} European conference on machine learning (ECML 1998), April 21–24, 1998* (pp. 4–15). Chemnitz, Germany.

Liu, H. (2004). MontyLingua: An end-to-end natural language processor with common sense. Available at < *http : //web.media.mit.edu/ hugo/montylingua* > (accessed 1 February 2005).

Manber, U. & Myers, G. (1990). Suffix arrays: a new method for on-line string searches. In *Proceedings of the first annual ACM SIAM symposium on discrete algorithms (SODA 1990), January 22–24, 1990.* San Francisco, California.

Market-Sentinel (2007). Market Sentinel. <www.marketsentinel.com> (accessed 4 October 2007).

Mettrop, W. & Nieuwenhuysen, P. (2001). Internet search engines: Fluctuations in document accessibility. *Journal of Documentation*, 57(5), 623–651.

Miller, G. A. (1995). WordNet: a lexical database for english. *Communications of the ACM*, 38(11), 39–41.

MySpace (2007). MySpace. <www.myspace.com> (accessed April 2007).

Nasukawa, T. & Yi, J. (2003). Sentiment analysis: capturing favorability using natural language processing. In *Proceedings of the 2^{nd} international conference on Knowledge capture, October 23–25, 2003.* (pp. 70–77). Florida, USA.

Pang, B. (2007). Polarity data set v2.0. October 1997, <http://www.cs.cornell.edu/people/pabo/movie-review-data/> (accessed 4 August 2007).

Pang, B. & Lee, L. (2004). A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42^{nd} annual meeting of the Association for Computational Linguistics (ACL), July 21–26 , 2004* (pp. 271–278). Barcelona, Spain.

Pang, B. & Lee, L. (2005). Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43^{rd} annual meeting of the Association for Computational Linguistics (ACL), June 25–30 , 2005* (pp. 115–124). University of Michigan, USA.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up ? sentiment classification using machine learning techniques. In *Proceeding of the conference on empirical methods in natural language processing (EMNLP 2002), July 6–7, 2002* (pp. 79–86). Philadelphia, PA, USA.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1–47.

Stone, P. J., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. (1966). *The general inquirer: a computer approach to content analysis*. The MIT Press.

Swan, R. & Allan, J. (2000). Automatic generation of overview timelines. In E. Yannakoudakis, N. J. Belkin, M.-K. Leong, & P. Ingwersen (Eds.), *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, July 24–28, 2000* (pp. 49–56). Athens, Greece.

Thelwall, M. (2000). Web impact factors and search engine coverage. *Journal of Documentation*, 56(2), 185–189.

Thelwall, M. (2008). Fk yea I swear: Cursing and gender in a corpus of MySpace pages. *Corpora*, 3(1), 83–107.

Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL), July 6–12 , 2002* (pp. 417–424). Philadelphia, PA, USA.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceeding of the conference on empirical methods in natural language processing (EMNLP 2005), October 6–8, 2005* (pp. 347–354). Vancouver, B.C., Canada.

Witten, I. H. & Frank, E. (2005). *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1–2), 69–90.

Yang, Y. & Liu, X. (1999). A re-examination of text categorization methods. In M. A. Hearst, F. Gey, & R. Tong (Eds.), *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, August 15–19, 1999* (pp. 42–49). Berkeley, USA.

Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the 14th international conference on machine learning (ICML 1997), July 8–12, 1997* (pp. 412–420). Nashville, Tennessee.

Yi, J., Nasukawa, T., Niblack, W., & Bunescu, R. (2003). Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the 3rd IEEE international conference on data mining (ICDM 2003), November 19–22, 2003* (pp. 427–434). Florida, USA.

Yin, L. L. & Savio, D. (1996). Learned text categorization by backpropagation neural network. Master's thesis, Hong Kong University of Science and Technology.