Computational Intelligence, Volume 22, Number 2, 2006

SENTIMENT CLASSIFICATION OF MOVIE REVIEWS USING CONTEXTUAL VALENCE SHIFTERS

ALISTAIR KENNEDY AND DIANA INKPEN
School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, K1N 6N5, Canada

We present two methods for determining the sentiment expressed by a movie review. The semantic orientation of a review can be positive, negative, or neutral. We examine the effect of valence shifters on classifying the reviews. We examine three types of valence shifters: negations, intensifiers, and diminishers. Negations are used to reverse the semantic polarity of a particular term, while intensifiers and diminishers are used to increase and decrease, respectively, the degree to which a term is positive or negative. The first method classifies reviews based on the number of positive and negative terms they contain. We use the *General Inquirer* to identify positive and negative terms, as well as negation terms, intensifiers, and diminishers. We also use positive and negative terms from other sources, including a dictionary of synonym differences and a very large Web corpus. To compute corpus-based semantic orientation values of terms, we use their association scores with a small group of positive and negative terms. We show that extending the term-counting method with contextual valence shifters improves the accuracy of the classification. The second method uses a Machine Learning algorithm, Support Vector Machines. We start with unigram features and then add bigrams that consist of a valence shifter and another word. The accuracy of classification is very high, and the valence shifter bigrams slightly improve it. The features that contribute to the high accuracy are the words in the lists of positive and negative terms. Previous work focused on either the term-counting method or the Machine Learning method. We show that combining the two methods achieves better results than either method alone.

Key words: sentiment classification, semantic orientation, valence shifters, machine learning, evaluation.

1. INTRODUCTION

Documents can be categorized in various ways, for example, by subject, genre, or the sentiment expressed in the document. We focus on sentiment classification (into positive or negative opinions). One useful application of sentiment classification is in question answering. Cases where a user is asking an opinion question such as *What are the reasons for the U.S.-Iraq war?* will require the system to determine the perspective of the different sources, using sentiment classification (Yu and Hatzivassiloglou 2003; Stoyanov et al. 2004). Another application is text summarization. If a program can pick out the sentiment of a review, it can use it to label the review; this could be an important part of the process of summarizing reviews (Pang, Lee, and Vaithyanathan 2002).

Two approaches to classifying sentiment are compared in this article. The first approach is to count positive and negative terms in a review, where the review is considered positive if it contains more positive than negative terms, and negative if there are more negative terms. A review is neutral if it contains an equal number of positive and negative terms. Instead of having a strict equality for neutral reviews, we can allow a margin of several terms.

Positive and negative terms are initially taken from the *General Inquirer* (Stone et al. 1966) (hereafter GI). GI is a dictionary that contains information about English word senses, including tags that label them as positive, negative, negation, overstatement, or understatement.

An enhanced term-counting method also takes contextual valence shifters into account. Valence shifters are terms that can change the semantic orientation of another term, for example, they make a positive term become negative. Examples of negation terms are *not*, *never*, *none*, *nobody*. There are many other factors that affect whether a particular term is positive or negative, depending on how it is used in a sentence, as shown in Polanyi and Zaenen (2004); however, we do not address all of them. Terms that change the intensity of a positive or negative term are also examined. These terms increase or decrease the weight

of a positive or negative term. We also add positive and negative terms from several other sources and test their contribution to the accuracy of the classification.

The second approach uses Machine Learning (ML) to determine the sentiment of the reviews. We trained Support Vector Machine (SVM) classifiers that use unigrams (single words) as features.

An enhanced version of this method uses as features, in addition to unigrams, some specific bigrams. We selected only bigrams that contain a combination of a negation, intensifier, or diminisher with another feature word. Rather than having bigrams such as *very good* where *very* is an intensifier, we identify the bigram as *int_good* where *int* indicates any intensifier. There are similar features for diminishers and negations. This is done to capture the type of the valence shifter. The two words in a bigram do not have to be right beside each other in the sentence. We used a parser to determine which negations/intensifiers/diminishers apply to which terms.

We note that the term-counting method has the advantage that it does not require training; thus it can be applied to reviews where training data are not available. The term-counting method is easily modified to include the linguistic analysis of Polanyi and Zaenen (2004). We can directly measure the impact of valence shifters on sentiment classification.

Methods based on ML are much more effective in terms of the accuracy of classification. With ML algorithms it is more difficult to show improvements by incorporating valence shifters, because they are already included, to some degree, in the basic classifier. Even when the classifier uses only unigrams as features, combinations of features detected by the ML algorithm can capture some aspects of the valence shifters. This can happen when combinations of terms, including valence shifters appear regularly in one class of documents (although not necessarily adjacent to each other). Because all terms in a document will affect its classification, valence shifters might have already been considered in classification.

We also combined the term-counting method and ML method. To do this, we needed predictions accuracy scores for the two methods. The two ways we used to combine the scores are: a simple weighted average of the two scores, and a meta-classifier that uses the scores as features. By combining the two methods we are able to improve the results over either of the method alone.

2. BACKGROUND AND RELATED WORK

Sentiment classification of reviews has been the focus of recent research. It has been attempted in different domains such as movie reviews, product reviews, and customer feedback reviews (Pang et al. 2002; Turney and Littman 2003; Pang and Lee 2004; Beineke, Hastie, and Vaithyanathan 2004; Gamon 2004). Much of the research until now has focused on training ML algorithms such as SVMs to classify reviews. Research has also been done on positive/negative term-counting methods and automatically determining if a term is positive or negative (Turney and Littman 2002).

2.1. Determining Sentiment

Research on predicting the semantic orientation of adjectives was initiated by Hatzivassiloglou and McKeown (1997). An unsupervised learning algorithm was used in Turney (2002) and Turney and Littman (2003) to determine the semantic orientation of individual terms. The algorithm started with seven known positive terms and seven known negative terms. The algorithm took a search term and used AltaVista's NEAR operator to find how many documents have the search term near the seven positive terms and the seven

negative terms. The difference in pointwise mutual information (PMI) score with the two sets was then used to determine the semantic orientation from pointwise mutual information (SO-PMI) score, which gives the degree to which each term is positive or negative (Turney and Littman 2002). The PMI score of two words w_1 and w_2 is given by the probability of the two words occurring together divided by the probabilities of each word in part:

$$\text{PMI}(w_1, w_2) = \log \frac{P(w1, w2)}{P(w_1)P(w_2)} = \log \frac{hits(w_1, w_2)N}{hits(w_1)hits(w_2)}$$

The formula for the semantic orientation of a word can be expressed as:

$$\text{SO-PMI}(word) = PMI(word, p\_query) - PMI(word, n\_query),$$

where the positive and negative reference terms are

$$\begin{aligned} p\_query &= good \text{ OR } nice \text{ OR } excellent \text{ OR } positive \text{ OR } fortunate \text{ OR } correct \\ &\quad \text{OR } superior \\ n\_query &= bad \text{ OR } nasty \text{ OR } poor \text{ OR } negative \text{ OR } unfortunate \text{ OR } wrong \\ &\quad \text{OR } inferior. \end{aligned}$$

OR and NEAR are operators offered by the AltaVista search engine (NEAR is no longer supported). By approximating the PMI values using number of hits returned by the search engine and ignoring the number of documents in the corpus (N), the formula becomes

$$\text{SO-PMI}(word) = \log \frac{hits(word \text{ NEAR } p\_query)hits(n\_query)}{hits(word \text{ NEAR } n\_query)hits(p\_query)}.$$

The semantic orientation of bigrams can also be determined (Turney 2002). The semantic orientation of terms and phrases can be used to determine the sentiment of complete sentences and reviews. Four hundred ten reviews from epinions.com were taken and the accuracy of classifying the documents was found when computing the sentiment of phrases for different kinds of reviews. Results ranged from 84% for automobile reviews to as low as 66% for movie reviews (Turney 2002).

2.2. Machine Learning for Determining Sentiment

One of the most common methods of classifying documents into positive and negative terms is to train an ML algorithm to classify the documents. Several ML algorithms are compared in Pang et al. (2002) and Pang and Lee (2004), where it was found that SVMs generally gave better results than other classifiers. Unigrams, bigrams, part of speech information, and the position of the terms in the text were used as features; however, using only unigrams was found to give the best results. Bayesian belief networks have also been used to determine the sentiment of a document (Bai, Padman, and Airoldi 2004).

Sentiment classification has also been done on customer feedback reviews (Gamon 2004). A variety of features were used in SVMs in an attempt to divide the data set not only into positive and negative, but to give rankings of 1, 2, 3, or 4, where 1 means "not satisfied" and 4 means "very satisfied." The proposed system was fairly good at distinguishing classes 1 and 4, with about 78% accuracy. Separating classes 1 and 2 from 3 and 4 was more difficult and was only 69% accurate. These results were achieved when using the top 2,000 features selected by log-likelihood ratios.

2.3. Distinguishing Objective from Subjective Statements

Methods for extracting subjective expressions from corpora are presented in Wiebe et al. (2004). Subjectivity clues include low-frequency words, collocations, and adjectives and verbs identified using distributional similarity. In Riloff and Wiebe (2003) a bootstrapping process learns linguistically rich extraction patterns for subjective expressions. High-precision classifiers label unannotated data to automatically create a large training set, which is then given to an extraction pattern learning algorithm. The learned patterns are then used to identify more subjective sentences.

A method of distinguishing objective statements from subjective statements is presented in Pang and Lee (2004). This method is based on the assumption that objective and subjective sentences are more likely to appear in groups. First, each sentence is given a score indicating if the sentence is more likely to be subjective or objective using a Naïve Bayes classifier trained on a manually annotated subjectivity data set. The system then adjusts the subjectivity of a sentence based on how close it is to other subjective/objective sentences. Although any improvements found using this method were not statistically significant, it was shown that you could reduce the size of a document, by removing objective sentences without decreasing the quality of the sentiment classification.

A similar experiment is presented in Yu and Hatzivassiloglou (2003). A Naïve Bayes classifier is used to discover opinion sentences by training it on a labeled data set. They also combine multiple Naïve Bayes classifiers for the same task, where each Naïve Bayes classifier focuses on a different part of the feature set. The feature sets included unigrams, bigrams, trigrams, part of speech information, and polarity. Once it was discovered whether a sentence is objective or subjective, a simple classifier (with unigrams as features) was used to determine the sentiment of the sentence.

3. THE DATA SET

We use a data set of classified movie reviews prepared by Pang and Lee (2004). This data set contains 2,000 movie reviews: 1,000 positive and 1,000 negative. A previous version of this data set, containing only 700 positive and 700 negative reviews, was used in Pang et al. (2002). The reviews were originally collected from the Internet Movie Database (IMDb) archive rec.arts.movies.reviews. Their classification as positive or negative is automatically extracted from the ratings, as specified by the original reviewer. They are currently available at http://www.cs.cornell.edu/people/pabo/movie-review-data/. Only reviews where the author indicated the movie's rating with either stars or some numerical system were included. No single author could have more than 20 reviews in the original data set.

4. METHODOLOGY

4.1. The Term-Counting Method

As we mentioned, the idea behind the term-counting method is simple: if there are more positive than negative terms then it is considered to be positive. If there are more negative than positive terms it is considered to be negative. If there are equal numbers of positive and negative terms it is neutral. We could have a margin instead of strict equality for neutral reviews; however, in our data set there are no neutral reviews. Therefore, in our case we prefer to have very few reviews classified as neutral. This idea of counting positive and negative

FUN#1	H4Lvd Positiv Pstv Pleasur Exprsv WlbPsyc WlbTot Noun PFREQ 97% noun-adj: Enjoyment, enjoyable
FUN#2	H4Lvd Negativ Ngtv Hostile ComForm SV RspLoss RspTot SUPV 3% idiom-verb: Make fun (of) – to tease, parody

FIGURE 1. GI entries for the word *fun*.

NOT	H4Lvd Negate NotLw LY — adv: Expresses negation NotLw LY adv: Expresses negation
FANTASTIC	H4Lvd Positiv Pstv Virtue Ovrst EVAL PosAff Modif — Virtue Ovrst EVAL PosAff Modif
BARELY	H4Lvd Undrst Quan If LY — Quan If LY

FIGURE 2. GI entries for the words *not*, *fantastic*, and *barely* (the tags Pstv and Ngtv are earlier versions of Positiv and Negativ in GI).

terms and expressions was proposed by Turney (2002). We augment this method by taking contextual valence shifters into account.

Identifying positive and negative terms: The main resource used for identifying positive and negative terms is the GI[1] (Stone et al. 1966). GI is a system that lists terms as well as different senses for the terms. For each sense it provides a short definition as well as other information about the term. This includes tags that label the term as being positive, negative, a negation term, an overstatement, or an understatement. The labels are for each sense of a word. For example, there are two senses of the word *fun* as seen in Figure 1. One sense is a noun or adjective for *enjoyment* or *enjoyable*. The second sense is a verb that means *to ridicule or tease*, *to make fun of* . The first sense of the word is positive, while the second is negative. The entry also indicates that the first sense is more frequent than the second sense (estimated to occur 97% of the times while the second sense occurs only 3% of the times).

We also examine negations, intensifiers, and diminishers. In GI intensifiers are known as *overstatements* and diminishers are known as *understatements*. Figure 2 shows the GI entries of the words *not*, *fantastic*, and *barely*, which are examples of a *negation*, an *overstatement*, and an *understatement*, respectively.

The GI contains 1,915 positive senses and 2,291 negative senses. We add more positive and negative senses from *Choose the Right Word* (Hayakawa 1994) (hereafter CTRW). CTRW is a dictionary of synonyms, which lists nuances of lexical meaning, extracted by Inkpen et al. (2005). After adding them, we obtain 1,955 positive senses and 2,398 negative senses. An example of a negative term from CTRW is *smugness*, while a positive example is *soothing*. Both of these terms are not found in GI. There are 696 overstatements and 319 understatements in GI. When we add those from CTRW, we obtain 1,269 overstatements and 412 understatements.

We also add positive and negative terms from other sources: a list of positive and negative adjectives (Adj) (Taboada and Grieve 2004), and a Web corpus (more details about the corpus are given in Section 5.1). In this list of adjectives we find negative terms such as *whiney* and positive terms such as *trendy*. In the corpus, we use SO-PMI scores to discover new positive and negative terms. Names such as *Hitler* and *Saddam* are found to be negative using SO-PMI. An example of a positive word found using SO-PMI is *happily*, which does not appear in GI, CRTW, or the list of adjectives.

[1] http://www.wjh.harvard.edu/∼inquirer/homecat.htm.

Lemmatization: The method of counting positive and negative terms requires us to transform the terms into their base forms (lemmas) to be able to check if a term is in our list of terms. The terms from GI are all in base form, there are no plurals and other inflected forms. When computing semantic orientation from a corpus, we can keep the terms in their inflected forms, or we can lemmatize them. For the ML experiments presented in Section 4.3, we lemmatized the terms to reduce the number of features used in classification. There are many lemmatizers available. We used the lemmatizer incorporated into the Xerox Incremental Parser (XIP) (Ait-Mokhtar, Chanod, and Roux 2002).

Word sense disambiguation: There are many terms that have multiple meanings. These terms have multiple definitions in GI. If a term is found for which there are many different definitions we may need to find out which definition corresponds to the correct sense. Often if one sense of a term is positive/negative, the other senses of the term will also be positive/negative. In GI there are only 15 words for which there are both positive and negative senses and only 12 words for which there are both intensifier and diminisher senses. After adding terms from CTRW there are 19 words that have both positive and negative senses and 37 words that have both intensifier and diminisher senses. We attempt two methods to determine which label to use for each word, without doing word sense disambiguation. In the first method we simply take all the senses and sum the number of senses that are positive and negative. If there are more positive senses than negative we consider the term positive, if there are more negative senses than positive, we consider it negative, and if there is an equal number, or no positive/negative senses, then it is considered neutral. The second method is to select the label of the sense that is estimated to be the most frequent, as listed by GI. We found that the second method works better; therefore, the results presented in Section 5 are for this case.

4.2. Incorporating Valence Shifters into the Term-Counting Method

There are two different aspects of valence shifting that are used to extend our term-counting method. First, we take into account negations that can switch the sentiment of positive or negative terms in a sentence. Second, we take intensifiers and diminishers into account.

Negations: Negations are terms that reverse the sentiment of a certain word (Polanyi and Zaenen 2004). For example consider the sentence *This movie is good* versus *This movie is **not** good*. In the first one *good* is a positive term and thus this sentence is positive. When *not* is applied to the clause, *good* is being used in a negative context and thus the sentence is negative (Polanyi and Zaenen 2004). We parsed the texts using the Xerox parser to obtain the scope of the negations.

Intensifiers and diminishers: Intensifiers and diminishers are terms that change the degree of the expressed sentiment. For example, in the sentence *This movie is **very** good*, the phrase *very good* is more positive than just *good* alone. Another example of an intensifier is *deeply* from the phrase ***deeply** suspicious*, which increases the intensity of the word *suspicious* (Polanyi and Zaenen 2004). On another side, in the sentence *This movie is **barely** any good*, the term *barely* is a diminisher, which makes this statement less positive. Another term that decreases the intensity of a phrase is *rather* from the phrase ***rather** efficient* (Polanyi and Zaenen 2004). These are examples of *overstatements* and *understatements* from GI. Overstatements are intensifiers, which increase the intensity of a positive/negative term, while understatements are diminishers, which decrease the intensity of that term. We note that the word *understatement* has other uses in linguistics (it could mean an entire clause or phrase). Here we use it to mean a *diminisher* term. To allow for intensifiers and diminishers

all positive sentiment terms in our system are given a value of 2. If they are preceded by an intensifier in the same clause then they are given a value of 3. If they are preceded by a diminisher in the same clause then they are given a value of 1. Negative sentiment terms are given a value of -2 by default and -1 or -3 if preceded by a diminisher or an intensifier, respectively. These values were proposed by Polanyi and Zaenen (2004) in their linguistic analysis study. There are a few places where this system does not work quite as you might expect. For example, if you have the phrase *not very good* then the *not* and *very* will combine to give a value of -3, which is then multiplied by 2 (the value for *good*) to give -6. This is a more negative value than you probably want for the phrase *not very good*. We could extend the scoring method to deal with such cases, but they are very rare in our data set.

CTRW also contains a large number of terms, which are listed as having high strength or low strength. These strengths do not strictly mean that they are intensifiers or diminishers, but many of them can be used as such. We compared results when adding intensifiers from CTRW with only using the intensifiers from GI.

The exact scope of these modifiers was extracted from a parsed version of the reviews. The reviews were parsed with the Xerox parser. We limited the valence shifters to negations, intensifiers, and diminishers. There are other, more complex, valence shifters that could be implemented (Wilson, Wiebe, and Hoffmann 2005).

4.3. The Support Vector Machine Classifiers

ML approaches were shown to achieve good results on the task of sentiment classification. Pang et al. (2002) presented results on the movie review data set. The SVM classifier outperformed the Naïve Bayes (by 4 percentage points). On the earlier version of the data set, the SVM classifier outperformed the Naïve Bayes and the maximum entropy classifier (by 4 and 5 percentage points, respectively). Therefore, we decided to use only the SVM algorithm in our experiments. Other classifiers not only are likely to achieve lower accuracy, but most of them have difficulties in dealing with the very large number of features (up to 35,000). The features used by Pang et al. are unigrams and bigrams from the reviews.

SVM is a state-of-the-art supervised kernel method for ML. It was successfully adapted to text classification by Joachims (1999). The basic idea behind kernel methods is to embed the data into a suitable feature space F via a mapping function $\phi : X \rightarrow F$, and then use a linear algorithm for discovering nonlinear patterns. The kernel function K acts as an interface between the data and the learning algorithm. During the learning phase, a weight $\lambda_i \geq 0$ is assigned to any example $x_i \in X$. All the labeled instances x_i such that $\lambda_i > 0$ are called support vectors. The support vectors lie close to the best separating hyper-plane between positive and negative examples. New examples are then assigned to the class of its closest support vectors, according to equation $\phi(x) = \sum_{i=1}^{n} \lambda_i K(x_i, x) + \lambda_0$.

Our basic ML classifier uses unigrams as features. We try three different methods for selecting unigrams. The first method is to simply select all unigrams that occur more than three times in the data set, to eliminate very rare terms and spelling errors (Pang et al. (2002) did the same in their experiments). The second method is to use only the unigrams from the reviews that appear in GI as positive or negative terms. This has the advantage of drastically reducing the feature set. The third method for selecting unigrams is to use all the positive and negative terms from GI, CTRW, and Adj as the feature set. The second and third feature sets incorporate into the ML approach information from the lists of positive and negative terms. The values of the features are Boolean. The value is 1 if the feature word appears in the review to be classified, and 0 otherwise. We could have used as value the frequency of the word in the review, but Pang et al. showed that in this case the performance of SVM is worse than in the case of Boolean values.

Our enhanced ML classifier takes the three different feature sets that are described for the basic ML classifier and then incorporates particular bigrams. For each feature term w we add three bigrams: one for a negation of the terms, one for an intensifier, and one for a diminisher. These three bigrams are given Boolean values and are referred to as neg_w (negation), int_w (intensifier), and dim_w (diminisher). For example, if the term w has an intensifier applied to it then $int\_w = 1$, or if it had a negation and a diminisher applied to it then $neg\_w = 1$ and $dim\_w = 1$. If some of these feature bigrams do not occur in the data set, we do not use them as features.

4.4. Combining the Term-Counting Method and the Machine Learning Method

To combine the two approaches we consider the results of the two methods on each review. We need prediction accuracy scores for the two methods. For the term-counting methods we devised our own prediction accuracy score, as the difference between the number of positive terms and the number of negative terms divided by the total number of positive and negative terms in the review. For the ML methods a score is provided by SVM Light (Joachims 1999) for each document. This score is not a confidence score per se; it is the value of the decision function used by SVM (its sign—positive or negative—gives the class). While the score produced by SVM Light, does not have an absolute meaning, its relative values were used for ranking documents after classification (Joachims 2002). In our combined method we combine the term-counting score with the SVM score. We also combine the term-counting score with a binary SVM score ($+1$ or -1, to reflect the class selected by SVM).

We combine the scores of the two methods in two ways: by taking a weighted average of the scores for each document (we simply add the score of the term-counting method, multiplied by a weight, to the SVM score); by training a meta-classifier that uses the two prediction accuracy scores as features for another SVM classifier.

5. EXPERIMENTAL SETUP AND RESULTS

5.1. The Term-Counting Systems

Table 1 presents the results of the experiments for both a basic system and an enhanced system. The basic system simply counts positive and negative terms, while the enhanced system adds the treatment of contextual valence shifters. Several dictionaries and word lists were used in the experiments: the GI; additional positive and negative terms, and additional overstatements and understatements from CTRW; list of positive/negative adjectives (Adj); and a longer list of positive/negative terms (SO-PMI).

We look at the accuracy of the classification (it varies from 61% to 63.4% in Table 1), as well as the precision, recall, and F-measure for each class. The precision, recall, and F-measure show whether the loss in performance is for the positive or for the negative class.

In the table we first present results for the basic and the enhanced systems when using only the terms in GI. Next, we present the results of the basic and enhanced systems when more positive and negative terms and more intensifiers and diminishers are added from CTRW.

Then we present the results when positive and negative adjectives from (Taboada and Grieve 2004) are added. The 1,718 adjectives in this list come with semantic orientation scores based on hit counts collected through the AltaVista search engine. The scores are computed using Turney's method (Turney and Littman 2002) explained in Section 2.1. We determined two thresholds: terms with SO-PMI value below 1.1 were labeled as negative, terms rated above 1.7 were labeled as positive. To pick the thresholds we tried a variety of different thresholds and evaluated them by computing the accuracy of the classifier on a small

TABLE 1. Results for all Systems, for the Term-Counting Methods*

System	Class	Accuracy	Precision	Recall	F-Score
Basic: GI	Positive	0.611	0.599	0.798	0.684
	Negative		0.700	0.425	0.529
Basic: GI & CTRW	Positive	0.612	0.600	0.794	0.684
	Negative		0.699	0.430	0.533
Basic: GI & CTRW & Adj	Positive	0.665	0.667	0.693	0.680
	Negative		0.693	0.637	0.664
Basic: GI & SO-PMI 1	Positive	0.581	0.871	0.195	0.319
	Negative		0.551	0.966	0.702
Basic: GI & SO-PMI 2	Positive	0.619	0.595	0.833	0.687
	Negative		0.731	0.405	0.521
Enhanced: GI	Positive	0.628	0.604	0.785	0.683
	Negative		0.694	0.476	0.565
Enhanced: GI & CTRW	Positive	0.630	0.606	0.784	0.684
	Negative		0.694	0.476	0.565
Enhanced: GI & CTRW & Adj	Positive	0.678	0.673	0.701	0.687
	Negative		0.691	0.655	0.673
Enhanced: GI & SO-PMI 1	Positive	0.589	0.882	0.209	0.338
	Negative		0.552	0.969	0.703
Enhanced: GI & SO-PMI 2	Positive	0.634	0.606	0.832	0.701
	Negative		0.728	0.437	0.546

*The basic system counts positive and negative terms. The enhanced system adds contextual valence shifters. Various lists of terms are used.

sample of the data. The terms with scores between the two thresholds are considered neutral. In Table 1 this list of terms is denoted as Adj.

In the last two versions of the basic and enhanced system that we present in Table 1, we used a much longer list of positive and negative terms. We computed SO-PMI scores for all the 38,790 content words in our data sets. To determine the SO-PMI scores we also used Turney's method, but instead of using AltaVista's NEAR operator (which is no longer available) we used the Waterloo MultiText System with a corpus of about one terabyte of text gathered by a Web crawler (Clarke and Terra 2003). We collected co-occurrence counts in a window of 20 words. The formula is similar to the one from Section 2.1:

$$\text{SO-PMI}(word) = \log \frac{hits([20] > word\ ..\ p\_query)hits(n\_query)}{hits([20] > word\ ..\ n\_query)hits(p\_query)}$$

except that the NEAR operator is replaced with counts in a window of 20 words.

After we computed the SO-PMI scores, we used the positive/negative terms from GI to automatically determine the best thresholds for the positive and negative terms. Terms with scores greater than 0.818 are positive, while terms with values less than -0.1845 are negative. This method gave a list of 4,357 positive terms and 12,633 negative terms, referred to as SO-PMI 1 in Table 1. We also tested this method with thresholds of 0.818 and -0.818, obtaining 4357 positive and 4291 negative terms—a more balanced ratio. This list is referred to as SO-PMI 2 in Table 1. We note that the positive/negative labels computed with SO-PMI are not always reliable. For example, when looking at the SO-PMI scores of the words from

TABLE 2. Results of the Machine Learning Experiments Using Leave-One-Out Cross-Validation

System	Class	Accuracy	Prec.	Recall	F-Score
Basic: SVM, GI unigrams	Positive	0.803	0.804	0.800	0.801
	Negative		0.801	0.806	0.803
Basic: SVM, GI&CTRW&Adj unigrams	Positive	0.820	0.823	0.815	0.818
	Negative		0.816	0.825	0.834
Basic: SVM, all unigrams	Positive	0.852	0.831	0.867	0.848
	Negative		0.873	0.837	0.854
Enhanced: SVM, GI unigrams + bigrams	Positive	0.811	0.814	0.805	0.809
	Negative		0.807	0.816	0.811
Enhanced: SVM, GI&CTRW&Adj unigrams	Positive	0.827	0.827	0.827	0.827
+ bigrams	Negative		0.827	0.827	0.827
Enhanced: SVM, All unigrams + bigrams	Positive	0.859	0.871	0.841	0.855
	Negative		0.846	0.876	0.860

our large list that are also in GI, the accuracy of labeling them is 65% for the best possible thresholds (0.818 and −0.1845).

We also ran tests that take negations into account, but not intensifiers or diminishers. For the movie reviews this method performed better than the basic system but worse than the enhanced system. The results of this system are not included in this paper. For example, we did not show in Table 1 the results without the additional intensifiers and diminishers from CTRW, because these results were nearly identical to the one shown for the versions GI and CTRW. Therefore, we can say that negation terms contribute a lot; however, when intensifiers and diminishers are added on top of negations their impact is not as large.

5.2. The Machine Learning Experiments

Table 2 presents the results of the SVM classifiers. They are obtained by leave-one-out cross-validation (the classifiers are trained on all examples except one; the testing is done on the left-out example; the process is repeated for all the examples in the data set). We used the tool SVM Light (Joachims 1999). The basic system uses only unigrams as features, while the enhanced system uses unigrams as well as certain bigrams containing a negation/intensifier/diminisher with a feature word.

The words used as unigram features are lemmatized to their base form. Lemmatizing eliminates about 5,500 features, while eliminating unigrams that occur three times or less eliminates about 19,000 additional features. We used three different sets of unigram features. The first of these unigram feature sets consists of the positive/negative terms from GI, a total of 3,066 features (words from GI that also appear in our corpus). The second feature set uses the positive/negative terms from GI&CTRW&Adj, which gives a total of 3,596 features. The last one uses all unigrams that appear more than three times, as described above. This leads to a total of 14,290 features. The enhanced system takes the unigram features and adds the valence shifting bigrams as features. When using GI there are a total of 7,534 features, for GI&CTRW&Adj there are 8,889 features and when adding bigrams to all unigrams there are 34,718 features. Bigrams were only added if they contained unigrams that existed in the corresponding unigram file. The total number of features for each system are summarized in Table 3.

TABLE 3. The Number of Features Used in Each Classifier

Feature Set	Number of Features
All unigrams	14,290
All unigrams + bigrams	34,718
GI unigrams	3,066
GI unigrams + bigrams	7,534
GI&CTRW&Adj unigrams	3,596
GI&CTRW&Adj unigrams + bigrams	8,889

The best accuracy, 85.9%, is achieved by the last system in Table 2. The improvement obtained by adding the valence shifting bigrams is small, but statistically significant[2] ($\alpha = 0.05$). It is remarkable that we were able to beat the very high baseline represented by the basic system. Pang et al. (2002) found that by adding all the bigrams as features, the SVM classifier performed worse than when using only unigrams. We have found that by selecting specific bigrams it is possible to improve over just unigrams. Pang et al. (2002) and Pang and Lee (2004) could not improve the results of SVM neither by adding bigrams nor by separating the objective part of the reviews.

5.3. Results of Combining the Two Methods

To test the combined method, we performed 10-fold cross-validation on the reviews data set. For each fold, we used 900 positive and 900 negative reviews for training and 100 positive and 100 negative reviews for testing. We computed the document scores for both the SVM and the term-counting methods. The term-counting method gives scores in the range from -1 to 1 for each document, while the SVM scores can range from as low as -2.7 to as high as 2.8 for our data.

In our weighted voting method, we multiplied the score of the term-counting method by a threshold and then we added the two classifiers scores for each document. We chose the threshold by testing various values and seeing which one maximizes the accuracy over 10 runs of the cross-validation (the optimal threshold was 0.65). We also trained an SVM meta-classifier that uses the scores of the two classifiers as features. Then we repeated the experiment by using a binary score for the original SVM classifier (to indicate the predicted class). The results of all these combination methods were very close to each other; therefore we report only the best results, for the SVM meta-classifier with non-binary scores.

Table 4 shows the results of the combined systems. First, we combine the basic term-counting system with the basic ML system. Then we combine the enhanced systems for term counting and the ML that uses valence shifter bigrams. We found that the basic term-counting method is 66.5% accurate, while the basic ML method is 84.9% accurate. When we combine the two methods we obtain an accuracy of 85.4%. For the enhanced system that uses valence shifters, we originally get 67.8% accuracy for the term-counting method and 85.5% for the ML method. When the two classifiers are combined together we get 86.2% accuracy, an increase of 0.7 percentage points. Although combining the systems improves the results,

[2] We performed statistical significance tests using the paired *t*-test, as described in Manning and Schütze (1999, p. 209).

TABLE 4. Results of Combining the Term-Counting and Machine Learning Experiments*

System	Class	Accuracy	Precision	Recall	F-Score
(1) Basic: Term counting, GI&CTRW&Adj	Positive	0.665	0.667	0.693	0.680
	Negative		0.693	0.637	0.664
(2) Basic: SVM, all unigrams	Positive	0.849	0.864	0.828	0.846
	Negative		0.835	0.870	0.852
Basic: Combined	Positive	0.854	0.858	0.848	0.853
	Negative		0.850	0.860	0.855
(1) Enhanced: Term counting, GI&CTRW&Adj	Positive	0.678	0.673	0.701	0.687
	Negative		0.691	0.655	0.673
(2) Enhanced: SVM, all unigrams + bigrams	Positive	0.855	0.871	0.834	0.852
	Negative		0.841	0.876	0.858
Enhanced: Combined	Positive	0.862	0.864	0.857	0.861
	Negative		0.858	0.866	0.862

*The results of the two SVM classifiers are repeated from Table 2 with the difference that here we report results of 10-fold cross-validation not leave-one-out cross-validation.

the improvement is not statistically significant. But the improvement from the system Basic: SVM, All Unigrams to the system Enhanced: Combined is significant, 1.3 percentage points.

By looking at the distribution of the errors in the 10-fold cross-validation experiments (Table 4), we see that the combined method has the potential to improve the results, because there are quite a few cases where SVM incorrectly classifies something, but the term-counting method classifies it correctly. When using unigrams as features, out of the 2,000 reviews, there were 302 misclassifications for SVM; of those, 149 were correctly classified by the term-counting method. SVM works worse for positive than for negative examples, getting 172 wrong, of which the term-counting method got 91 right. For negative examples, SVM misclassified 130 examples, of which the term-counting method correctly classified 58. When using unigrams and bigrams as features, there were 290 misclassifications by SVM, 163 of which were classified correctly by the term-counting method. For positive examples, SVM made 166 errors, of which 97 were correctly classified by term counting. For negative examples SVM made 124 errors, of which term counting got 66 right. It seems that the term-counting method correctly classifies about half of the mistakes made by SVM (while making many more mistakes itself). It also seems to do a better job for the positive reviews than for the negative reviews.

6. DISCUSSION OF THE RESULTS

6.1. The Effects of Valence Shifters on the Term-Counting System

Our first goal was to determine the effectiveness of adding contextual valence shifters to the simple method of counting positive and negative terms. From our experiments it is clear that the addition of valence shifters has an improving effect on the classification of reviews. It can be seen in Table 1 that the accuracy for both data sets with all dictionaries and word lists improves when contextual valence shifters are added. In most cases the F-measure also improves when contextual valence shifters are included.

To measure only the impact of the valence shifters, for the positive/negative term-counting method, we compare the basic system and the enhanced systems. The improvement is statistically significant in all cases. For example, the gain of 1.7 percentage points (from 61.1% to 62.8%) between Basic: GI and Enhanced: GI from Table 1 is statistically significant at the level $\alpha = 0.05$.

Our enhanced term-counting method worked relatively well in comparison to experiments done using human judges. In Pang et al. (2002) two humans selected less than 20 positive and negative words each to determine the sentiment of movie reviews. These two humans scored 58% and 64%, compared to our 67.8% for our best term-counting system with valence shifters. These results cannot be directly compared with ours, though, because they were tested on an earlier version of the data set; also we gathered polarity terms without any domain knowledge that the human-made lists contained.

6.2. The Effect of Adding More Positive and Negative Terms

Two other things that we examined are the effect of adding more positive and negative terms, as well as the effects of adding more intensifiers and diminishers. Adding positive and negative terms from CTRW generally improved the accuracy of the classification. This is true for both the basic and the enhanced system (with and without contextual valence shifters). Adding overstatements and understatements from CTRW did not make a difference.

When we added a large number of positive and negative terms with automatically computed SO-PMI values, the performance was not always better. The accuracy decreased, especially for SO-PMI 1, which has too many negative terms. For Basic: GI the accuracy of classification falls from 61.1% to 58.1% for Basic: GI & SO-PMI 1. This is probably due to the fact that the positive/negative labels computed with SO-PMI are not always reliable. It is not always the case that SO-PMI hurts the results though, because for Basic GI & SO-PMI 2 the accuracy improves to 61.9%. The performance is better when using GI & SO-PMI 2, and worse when using GI & SO-PMI 1 compared to using only GI, for both the basic and the enhanced systems.

6.3. Comparing the Systems

Our enhanced term-counting system, in its best variant (Enhanced: GI&CTRW&Adj), achieved a statistically significant increase of 6.7 percentage points, compared to the baseline basic system (Basic: GI), improving from 61.1% to 67.8%.

The results found using ML are much higher, in the range of 80–85.9%. For every set of unigram features the results did improve slightly by adding our special bigrams. The initial classification accuracy for just unigram features was 85.2%, adding valence shifter bigrams improves it by just 0.7% to 85.9%. We also found that by using just known positive and negative terms as features we were able to obtain good results. By using only the positive/negative terms from GI as features we were able to get 80.3% accuracy. Results improved by using the positive/negative terms from GI&CTRW&Adj to 82.0%. These results were improved by between 0.7% and 0.8% by including valence shifter bigrams; this improvement is not statistically significant. The high accuracy obtained by the SVM classifiers that use only unigrams for the lists of positive/negative words as features, shows that the positive and negative terms have an important contribution to the success of the ML method. When we trained an SVM classifier using only the top most frequent 3,066 unigrams as features, the accuracy was high (84.4%); this is probably due to the fact that, in addition to the positive and negative terms, there are some other highly discriminate words (e.g., names of actors that usually appear in good movies).

We found that combining the two systems slightly improved the results. We were able to get an improvement of 1.3% by using only the SVM method by combining the document's scores from the SVM with those from the term-counting method. Despite the fact that term counting alone did not perform very well, it was able to give a considerable boost to the ML method, which already had results in the mid 80% range. The improvement could be due to the fact that the two classifiers do not make the same kind of classification errors.

Pang and Lee (2002) showed that adding all the bigrams can hurt the results of sentiment classification when using SVMs; however, we have shown that adding specific bigrams can actually help the results. We have also shown that the term-counting method together with the valence shifters can be used to improve the accuracy of a basic SVM classifier.

7. FUTURE WORK

There are many possible directions for future work. For example, we would like to explore the effect of using only subjective sentences in classifying reviews for both the term-counting methods and the ML method.

The positive and negative terms may not all be equally positive or negative. Positive and negative terms can be given weights (those could be, e.g., their SO-PMI scores) to show just how positive or negative they are. Overstatements and understatements could also be weighted.

Another way to improve the accuracy of classifying movie reviews could be to automatically build small domain models of salient objective key phrases. Positive and negative terms in these key phrases would be ignored in the term-counting method.

In the ML experiments, one direction of future research is to identify the named entities and to measure their contribution to the accuracy of the SVM classifiers. For example, it could be the case that some names of popular actors or directors appear most often in movies with positive reviews.

ACKNOWLEDGMENTS

We wish to thank Egidio Terra and Charlie Clarke for giving us permission to use the Waterloo MultiText System with the terabyte corpus of Web data, and Peter Turney and his colleagues at NRC/IIT for giving us access to their local copy of this system. Our research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Ottawa.

REFERENCES

AIT-MOKHTAR, S., J.-P. CHANOD, and C. ROUX. 2002. Robustness beyond shallowness: Incremental deep parsing. Natural Language Engineering, **8**(2–3):121–144.

BAI, X., R. PADMAN, and E. AIROLDI. 2004. Sentiment extraction from unstructured text using tabu search-enhanced Markov blanket. *In* Proceedings of the International Workshop on Mining for and from the Semantic Web, pp. 24–35, Seattle, Washington.

BEINEKE, P., T. HASTIE, and S. VAITHYANATHAN. 2004. The sentimental factor: Improving review classification via human-provided information. *In* Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), pp. 263–270, Barcelona, Spain.

CLARKE, C. L. A., and E. TERRA. 2003. Passage retrieval vs. document retrieval for factoid question answering. *In* Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 427–428, Toronto, Canada.

GAMON, M. 2004. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. *In* Proceedings the 20th International Conference on Computational Linguistics, pp. 841–847, Geneva, Switzerland.

HATZIVASSILOGLOU, V., and K. MCKEOWN. 1997. Predicting the semantic orientation of adjectives. *In* Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97), pp. 174–181, Madrid, Spain.

HAYAKAWA, S. I. (Editor). 1994. Choose the Right Word (2nd ed.), revised by Eugene Ehrlich, Harper Perennial, New York.

INKPEN, D. Z., O. FEIGUINA, and G. HIRST. 2005. Generating more-positive and more-negative text. *In* Computing Attitude and Affect in Text: Theory and Applications. *Edited by* J. Shanahan, Y. Qu, and J. Wiebe. The Information Retrieval Series, Vol. 20, Springer, Dordrecht, The Netherlands, pp. 187–196.

JOACHIMS, T. 1999. Making large-scale SVM learning practical. *In* Advances in Kernel Methods—Support Vector Learning. *Edited by* B. Schlkopf, C. Burges, and A. Smola. MIT Press, Cambridge, Massachusetts, pp. 169–184.

JOACHIMS, T. 2002. Optimizing search engines using clickthrough data. *In* Proceedings of the Eight ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 133–142, ACM Press, New York.

MANNING, C., and H. SCHÜTZE. 1999. Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts.

PANG, B., and L. LEE. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *In* Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), pp. 271–278, Barcelona, Spain.

PANG, B., L. LEE, and S. VAITHYANATHAN. 2002. Thumbs up? Sentiment classification using machine learning techniques. *In* Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, pp. 79–86, Philadelphia, Pennsylvania.

POLANYI, L., and A. ZAENEN. 2004. Contextual valence shifters. *In* Computing Attitude and Affect in Text: Theory and Applications. *Edited by* J. Shanahan, Y. Qu, and J. Wiebe. The Information Retrieval Series, Vol. 20, Springer, Dordrecht, The Netherlands, pp. 1–9.

RILOFF, E., and J. WIEBE. 2003. Learning extraction patterns for subjective expressions. *In* Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 105–112, Sapporo, Japan.

STONE, P. J., D. C. DUNPHY, M. S. SMITH, et al. 1966. The General Inquirer: A Computer Approach to Content Analysis. MIT Press, Cambridge, Massachusetts.

STOYANOV, V., C. CARDIE, D. LITMAN, and J. WIEBE. 2004. Evaluating an opinion annotation scheme using a new multi-perspective question and answer corpus. *In* Computing Attitude and Affect in Text: Theory and Applications. *Edited by* J. Shanahan, Y. Qu, and J. Wiebe. The Information Retrieval Series, Vol. 20, Springer, Dordrecht, The Netherlands, pp. 77–89.

TABOADA, M., and J. GRIEVE. 2004. Analyzing appraisal automatically. *In* Proceedings of the AAAI Symposium on Exploring Attitude and Affect in Text: Theories and Applications (published as AAAI technical report SS-04-07), pp. 158–161, Stanford, California.

TURNEY, P. 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 2002), pp. 417–424, Philadelphia, Pennsylvania.

TURNEY, P., and M. LITTMAN. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report ERB-1094, National Research Council, Institute for Information Technology.

TURNEY, P., and M. LITTMAN. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS), **21**(4):315–346.

WIEBE, J., T. WILSON, R. BRUCE, M. BELL, and M. MARTIN. 2004. Learning subjective language. Computational Linguistics, **30**(3):277–308.

WILSON, T., J. WIEBE, and P. HOFFMANN. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. *In* Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), pp. 347–354, Vancouver, Canada.

YU, H., and V. HATZIVASSILOGLOU. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *In* Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 129–136, Sapporo, Japan.