Incorporating Semantic Knowledge for Sentiment Analysis

Shitanshu Verma
IIT Bombay
Mumbai, India
shitanshu@cse.iitb.ac.in

Pushpak Bhattacharyya
IIT Bombay
Mumbai, India
pb@cse.iitb.ac.in

Abstract

We report work on using knowledge of sentiment-bearing words in statistical approaches to automatic sentiment analysis and opinion mining (SA & OM). Our main contribution lies in constructing document feature vectors that are *sentiment-sensitive* and use word knowledge. This is achieved by incorporating sentiment-bearing words as features in document vectors, extracted with the help of SentiWordNet which is essentially the wordnet with sentiment scores attached to the synsets. Support Vector Machines (SVM) have been used to classify documents into positive and negative polarity (*i.e.,* sentiment) classes. Experiments show that we achieve state of art performance in sentiment analysis of standard movie reviews dataset and locally created product review dataset.

1 Introduction

Sentiment analysis aims to categorize text as positive or negative on the basis of the positive or negative sentiment (opinion) expressed in the document towards a topic. A document with positive or negative sentiment is also said to be of positive or negative *polarity* respectively.

The granularity of the polarity can be up to the level of words. That is, there can be polar (subjective) and non-polar (objective/neutral) sentences and words.

There are several challenges in the task of sentiment analysis. **Firstly**, we have to do *subjectivity detection*, *i.e.*, selecting opinion containing sentences (Pang and Lee, 2004). Consider, for example, two sentences in a review of the city of Singapore. “*Singapore’s economy is heavily dependent on tourism and IT industry. It is an excellent place to live in.”* The first sentence is an objective or factual one and does not convey any sentiment towards Singapore. Hence this should not play any role in deciding on the polarity of the review, and should be filtered out. **Secondly**, Word Sense Disambiguation (WSD), a classical NLP problem is often encountered. For example, *“an unpredictable plot in the movie”* is a positive phrase, while *“an unpredictable steering wheel”* is a negative one. The opinion word *unpredictable* is used in different senses, *viz.*, *with-twists-and-turns* and *erratic* respectively. **Thirdly**, the problem of sentiment analysis has to grapple with thwarting, *i.e.*, sudden deviation from positive to negative polarity, as in “*The movie has a great cast, superb storyline and spectacular photography; the director has managed to make a mess of the whole thing”*. **Fourthly**, negations- in all its subtle and gross forms- unless handled properly can completely mislead. “*Not only do I not approve Lemon MX, but also hesitate to call it a radio”* has a positive polarity word *approve*; but its effect is negated by many negations. **Fifthly**, keeping the target in focus can be a challenge. Consider the following statement: “*my camera compares nothing to John’s camera which is sleek and light, produces life like pictures and is inexpensive”*. All the positive words about John’s camera being the constituents of the document vector will produce an overall decision of positive polarity which is wrong.

Our main contribution in the work lies in introducing *word level sentiment into the feature vector of the document*. Scores obtained from **SentiWordNet** which is essentially the English wordnet[1] with polarity scores attached to the synsets are used for providing weights to the features, *i.e.*, words of a document. This leads to better quality sentiment classification of documents.

The paper is structured as follows. Section 2 is on related work. Section 3 introduces the critical resource SentiWordNet used in our work. Section 4 is on feature engineering. Section 5 describes the corpora used for evaluation. Section 6 contains experiments, results and discussions. Finally in section 7, we present conclusions and future directions.

[1] http://princeton.wordnet.edu

Proceedings of ICON-2008: 6th International Conference on Natural Language Processing
Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2008

2 Related Work

Both statistical and rule based methods have been used for the polarity detection of a document. Pang and Lee (2002) use different classifiers based on unigram and bigram word vectors of documents. Turney (2002), on the other hand, uses semantic orientation for classification. Pang and Lee (2004) and Agarwal and Bhattacharyya (2005) have used graph cut based method to classify movie reviews. The minimum cut methods along with SVMs has been effective for polarity detection.

Of late, lexical resources and lexical resources based sentiment analysis have received attention. SentiWordNet (Esuli and Sebastiani, 2006) and WordNet Affect (Strapparava and Valitutt, 2004) are two such lexical resources.

Our work can be looked upon as falling in the line of appropriate feature selection for SA. A very recent work by Kim, Li and Lee (2009) models term weighting into a sentiment analysis system utilizing collection statistics, contextual and topic related characteristics as well as opinion related properties.

Work is going on in understanding how syntactic structures can influence sentiments. Ramanathan, Liu and Choudhary (2009) report work on sentiment analysis of conditional sentences.

3 A critical resource: SentiWordNet

SentiWordNet (Esuli and Sebastiani, 2006) is a lexical resource based on the English wordnet which incorporates sentiment information for each synset. For every synset *S* in wordnet three numerical scores *Pos(S), Neg(S)* and *Obj(s)* are calculated, describing the *positivity,* the *negativity* and the *objectivity* of the synset.

Consider from (Esuli and Sebastiani, 2006) the synset ***[estimable(3)]***, *i.e.*, the 3rd sense of the word *estimable* as given the English wordnet version 3.0. This corresponds to the sense "*may be computed or estimated*". The sense has an *Obj* score of 1.0, and *Pos* and *Neg* scores of 0.0. The synset ***[estimable(1)]*** corresponding to the sense "*deserving of respect or high regard*" has a *Pos* score of 0.75, a *Neg* score of 0.0, and an *Obj* score of 0.25.

These scores are based on the classification of the synset by a committee of classifiers (Esuli and Sebastiani, 2006a). For each synset, these scores range from 0.0 to 1.0 and **always sum up to 1**. Hence SentiWordNet can be used for polarity identification as well as for subjectivity detection.

Thus in the example of ***[estimable (1)]****,* 6 out of 8 classifiers judged it as *Pos*, none as *Neg* and 2 as *obj,* thus producing the score of 0.75, 0.0 and 0.25 respectively.

4 Feature engineering

The core component of our system whose architecture diagram is shown in figure 1 is feature engineering. We proceed to discuss this in detail.

First, subjectivity detection (details in section 6.1) and negation handling are done on the reviews. Sentiment score based pruning is then applied to this result. TF-IDF vectors are created at this stage, which are then subject to information gain based pruning. **The feature vector at this point consists mainly of opinion words**. An SVM classifier is applied on these vectors, which classifies documents in to positive and negative classes.

4.1 Handling Negation words

We consider three negation words *no*, *not* and *never.* The first word after the negation word, that is not a stop word, is taken as the negated word. For example, in the comment "*this could not be a good camera*", the word whose sense is reversed is *good* and it appears after two stop words *be* and *a*.

To incorporate negation information, the negation word and negated word are joined with a hyphen. For example *"not good"* is replaced with *not_good*. The sentiment score of this new "word" is the negative of the sentiment score of the negated word.

4.2 Details of feature engineering

First a set of standard preprocessing steps are carried out, *viz.*, *tokenizing, stemming* and *stop word removal*. Tools provided by *Rapidminer's text plugin*[2] were used for these tasks. Stemming was done by wordnet's morphological analyzer. After this, feature pruning is done two stages.

> *First, a sentiment score based pruning removes all non-opinion words. Following this, information gain based pruning is done to remove domain specific stop words and noisy words.*

These steps are discussed in detail in the following sections:

[2] http://rapid-i.com/content/blogcategory/38/69/

4.2.1 Sentiment score based pruning

A scoring function is used to calculate the sentiment score of all the words in the document using SentiWordNet. The score so obtained is compared with a threshold. Only words with score higher than the threshold are accepted. The threshold value is selected with care, such that, some characteristic words like *good, nice, boring etc.* are included.

This brings us to the problem of grappling with **word sense disambiguation (WSD)**. Given a word *W* in the document, which sense should we choose to pick up a SentiWordNet score? We use the three strategies described below:

I. Average of Max of *Pos* and *Neg*

$Score(W)= [\Sigma_K Max(Pos(W_K), Neg(W_K)]/K$

where,

$Pos(W_k)$	=	*Positive score given by SentiWordNet to the* k^{th} *Sense of W*
$Neg(W_k)$	=	*Corresponding negative score*
K	=	*Number of senses of word*

The intuition behind this score is that we have to choose for *W* one of *Pos, Neg* and *Obj* scores, as recorded for all its senses. *Obj* does not contribute to the sentiment value. Between *Pos* and *Neg*, we choose the greater value. After that we average over the sentiment scores of all senses and the score effectively becomes the *expected sentiment score* of a word.

II. Maximum over all the senses

$Score(W)=Max_K[Max(Pos(W_K), Neg(W_K))]$

This gives importance to the sense with the maximum polarity score. The underlying assumption is that people express their sentiments strongly and choose words with high polarity value.

III. Weighted average of all the senses

We consider two factors. (1) In the wordnet, words in a synset are arranged in the order of their frequency of use in that particular sense. (2) Synsets differ in the number of words contained in them.

The score that every sense contributes can be scaled by the position on the word in the synset and the number of words in the synset. This gives rise to the following formula for word scoring:

$$Score(W) = \frac{\sum_k weight_{k*} \max(pos(w_k), neg(w_k))}{\sum_k weight_k}$$

where,

$weight_k$ is the significance of the word in k^{th} synset. The weight is given by

$$weight_k = 1 - \frac{position\ of\ W\ in\ k^{th}\ synset}{number\ of\ words\ in\ k^{th}\ synset}$$

Position of *W* starts from 0. That is why for a word in the first position which is the position of the most frequent occurrence, *weight= 1.*

4.2.2 Information gain based pruning (IGBP)

Information gain is used to measure the importance of an attribute/feature (*X*) with respect to the class attribute (*Y*). Formally, information gain of an attribute/feature *X* with respect to a class attribute *Y* is the reduction in uncertainty about the value of *Y* when we know the value of *X*.

$$\boldsymbol{InfoGain}(\boldsymbol{Y};\boldsymbol{X}) = \boldsymbol{entropy}(\boldsymbol{Y}) - \boldsymbol{entropy}(\boldsymbol{Y}|\boldsymbol{X})$$

where *X* and *Y* are discrete variables taking values $\{x_1, x_2...x_m\}$ and $\{y_1,y_2,...y_n\}$ respectively. The entropy(Y) is defined as

$$\boldsymbol{Entropy}(\boldsymbol{Y}) = -\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{P}(\boldsymbol{Y} = \boldsymbol{y_i})\, \boldsymbol{log_2}\, \boldsymbol{P}(\boldsymbol{Y} = \boldsymbol{y_i})$$

The conditional entropy of *Y* given *X* is defined as

$$\boldsymbol{Entropy}(\boldsymbol{Y}|\boldsymbol{X}) = -\sum_{\boldsymbol{j=1}}^{\boldsymbol{m}} \boldsymbol{P}(\boldsymbol{X} = \boldsymbol{x_j})\boldsymbol{Entropy}(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x_j})$$

High *Information Gain* features reduce the uncertainty about the class to the maximum. In our document vectors we retain only those features, *i.e.*, words which cross an Information Gain threshold.

Information Gain based pruning is used after sentiment score based pruning. Features (words) like *graphic, find, seem etc.* appear in both positive and negative documents and do not play any role in deciding the sentiment of a document. By introducing an information gain based filter we remove these words/features.

4.3 Vector creation

Finally, a document vector is created through the stages depicted in Figure 2.

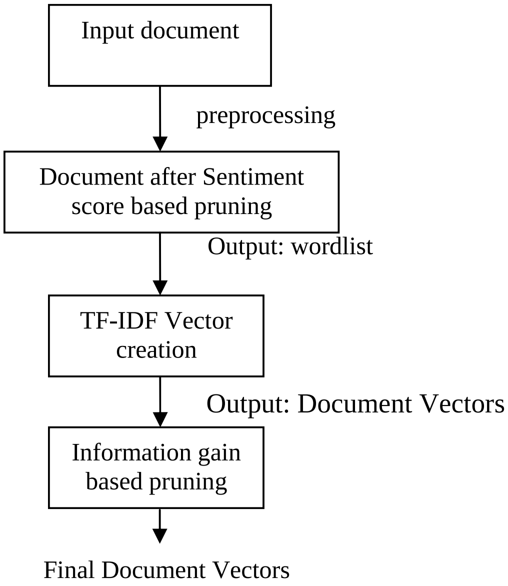

Fig 1: Information flow for document vector creation

5 Corpora used

We have performed experiments on three corpora: *movie review subjectivity corpus*, *movie review polarity corpus* and *product review corpus*. The **movie review subjectivity corpus** is provided by Pang and Lee with the paper they published in ACL 2004 (Pang and Lee, 2004). It contains 5000 subjective sentences and 5000 objective sentences. Each sentence is at least 10 words long. The subjective sentences were taken from movie review snippets from www.rottentomatoes.com. The objective sentences were taken from the plot summaries of movies from the Internet Movie Database (www.imdb.com). Review snippets from *rottentomatoes.com* were assigned *subjective class* while *imdb* plot summaries were assigned *objective class*.

The **Movie review polarity corpus** is provided by Pang and Lee with the paper they published in EMNLP 2002 (Pang, *et al.*, 2002). It contains 1000 positive movie reviews and 1000 negative movie reviews. The reviews are collected from Internet Movie Database archive of *rec.arts.movies.reviews* newsgroup. Only those reviews were selected where some ratings were given. The ratings were either in the form of stars or numerical score. The ratings were automatically processed to categorize reviews in three categories *positive, negative* and *objective*. Only positive and negative reviews were considered for our work.

The **product review corpus** was annotated by us for this work. It contains reviews on digital cameras and camcorders collected from www.reviews.cnet.com. The corpus contains 1100 positive reviews and 800 negative reviews. A typical review contains a review title, pros and cons about the product and a free text summary. We have used full reviews for our experiments. The corpus was labeled manually after reading the reviews. The label of the review was based on the rating given by the author, the summary given and overall satisfaction of the author.

Product review domain considerably differs from movie review domain because of two reasons. **Firstly**, there are *feature specific* comments in product reviews. People may like some features and dislike some others. Thus reviews consist of both positive and negative opinions, which make the task of classifying the review as positive or negative tougher. Such feature specific comments occur less frequently in movie reviews. **Secondly**, there are a lot of comparative sentences in product reviews and people talk about other products in reviews. This makes the task of *opinion target detection* an important aspect of the problem.

6 Experiments, Results and Discussions

We have performed 3 sets of experiments on 3 different corpora described above. The experiments were designed on *Rapidminer*[3]. Document vectors were created using the *Text Plugin of RapidMiner*. *LibSVM learner with a linear kernel* was used for classification purposes and all the results are obtained by 10 fold cross validation method.

6.1 Subjectivity detection: as a step towards polarity detection

These experiments were performed with the movie review subjectivity corpus. The text was first pre-processed using standard NLP techniques of tokenizing, stop word removal and stemming. The classifier was trained with different values of *C*, the cost parameter of SVM. It gave best results for *C=1*. The highest F-Measure obtained was **88.53%.** The classifier was then used to classify the sentences of a movie review corpus as *subjective* or *objective*.

6.2 Polarity detection

These experiments were performed on the movie review polarity corpus. First document vectors

[3] http://rapid-i.com/

were created and then these document vectors were classified with SVM classifier.

Although the number of features for the three scoring functions is quite close, the features selected were found to be quite different.

The feature vector obtained **after sentiment score based pruning** on full review contains many useless features like *world, reason, sound, role etc.* that do not contain any important information related to the sentiment of the document. These words were removed with **information gain based pruning** (IGBP). Domain specific stop words like *cast, hero, comedy etc.* are removed from the feature vector. This step helps a lot in classification.

Input	Accuracy of system (in %)					
	Average		Weighted Average		Maximum	
	Normal	Scaled	Normal	Scaled	Normal	Scaled
Full review	70.35	67.47	70.44	71.80	69.83	71.80
Full review after IGBP	74.47	74.5	78.1	77.83	77.83	78.9
Sub-jective review after IGBP	82.1	81.5	84.64	84.01	84.31	**85.61**

Table 1: Accuracy on movie review corpus

Table 1 shows that accuracy improves after every stage of processing, *viz.*, Information Gain Based Pruning and extracting Subjective sentences.

6.2.1 Comparison with state of the art

Our method outperform the method used by Pang and Lee in (Pang, et al., 2002), which applies machine learning techniques on unigram and bigram features. We achieve the highest accuracy of **85.61%** as compared to 82.9% in (Pang, et al., 2002), on the same corpus. The value also improves over the accuracy figure of 86.4% in (Pang and Lee, 2004) that uses min cut algorithm with SVM classifier

6.3 Product review corpus

After the sentiment score based pruning, some noisy words like *check, cover, flash, black, white etc.* still remain, which are removed in the next phase of Information gain based pruning. Opinion words like *fail, mediocre, trust, horrible etc*. get high scores, though.

Following are the results for product review corpus:

Input	Accuracy of system (in %)			
	Before IGBP		After IGBP	
	Unscaled	Scaled	Unscaled	Scaled
Average	77.40	76.45	77.71	77.60
Weighted Average	79.50	80.0	80.10	79.20
Maximum	80.45	81.80	82.65	**83.91**

Table 1: Accuracy for product review corpus

7 Error Analysis

Our method of SA & OM, like others in the field, is essentially a **bag-of-words** approach with its usual problems of *context insensitivity, focus drift etc*. Because of this, our error analysis of results reveals the following facts which are not unexpected:

(i) Contrast between expectation and opinion
Consider the review, "*I expected this movie to be very good. The casting was good, director was good but it crushed all my expectation.*" In such cases the feature vector consists of positive words and the review is classified as positive which is inappropriate.

(ii) Focusing on the part rather than the whole
Consider the following review, "*Steven Spielberg is a very good director. His movies have substance, they are well directed. But this movie was not up to the mark. The direction is so so, storyline is slow*" The review hardly has any negative words, but it talks about a general category of the movies (directed by Steven Spielberg). The actual sentiment towards the movie is mild and expressed with a minimal use of opinion words. Our system fails to classify such reviews correctly.

(iii) Target defocusing
In product review domain, almost 70% of misclassified reviews are negative ones. This is because of three reasons. *Firstly*, the author tends to compare the new camera (being reviewed) with their old camera or a friend's camera. They praise the other camera and discuss the positives of other camera. Since our method is based on a *bag of words* features, it fails to handle such situation. It just looks at the positive opinion words used by author to praise the other camera and

classifies the review as a positive review. Methods used in (Jindal, et al., 2006) (Ganapathibhotla, et al., 2008) can be applied to handle such comparative sentences. *Secondly*, the negative reviews that do not have comparative sentences are small. The document vector created is very sparse and many times does not contain enough information to correctly classify the review.

(iv) Technical documents

Technical reviews which contain a lot of statistics are misclassified. The system does not have any prior knowledge about the specifications of the product and it fails to infer the **implicit polarity** from the numbers.

8 Conclusion and Future work

We have built a sentiment analysis system which makes use of *word knowledge* provided by a rich lexical resource in the form of SentiWordNet to prepare more informative document vectors. On the standard data set of movie reviews, the system outperforms well reported techniques such as those in (Pang and lee, 2004; Pang, et al., 2002).

The system has also been tested on *product review* which to the best of our knowledge is an uncharted domain. The performance of the system on this domain is satisfactory except for low recall (70%) on negative polarity documents. Investigations on causes of errors points to the classic limitations inherent in bag-of-word based approaches.

On the feature extraction front, our system performs unexpctedly well on domain specific opinion words like *return, blurry* (product review domain) and *skip* (as in "*skipping a movie"* which expresses a negative opinion in the movie review domain). These features are not easy to detect.

Future work consists in deploying better subjectivity detection techniques, like graph cut methods. The interaction of WSD and SA is a fertile area of research (Wiebe and Mihalcea, 2006; Akkaya, et al., 2009). Scoring functions that incorporate the frequency of *use of a synset* should be explored. Also, methods used in (Jindal, et al., 2006) (Ganapathibhotla, et al., 2008) can be used to remove comparative sentences to improve performance of the system.

References

Alekh Agarwal and Pushpak Bhattacharyya. 2005. *Sentiment analysis: A New Approach for Effective Use of Linguistic Knowledge and Exploiting Similarities in a Set of Documents to be Classified.* ICON 05.

Cem Akkaya, Janyce Wiebe and Rada Mihalcea. 2009. *Subjectivity Word Sense Disambiguation.* Emperical Methods in Natural Language Processing.

Kushal Dave, Steve Lawrence and David M. Pennock. 2003. *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews.* WWW2003.

Andrea Esuli and Fabrizio Sebastiani. 2006. *SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining.* LREC-06.

Andrea Esuli and Fabrizio Sebastiani. 2006a. *Determining Term Subjectivity and Term Orientation for Opinion Mining.* European Chapter of the Association for Computational Linguistics EACL .

Murthy Ganapathibhotla and Bing Liu. 2008. *Mining Opinions in Comparative Sentences.* Coling-2008.

Nitin Jindal and Bing Liu. 2006. *Identifying Comparative Sentences in Text Documents.* ACM SIGIR-06.

Jaungi Kim, Jin-Ji Li and Jong-Hyeok Lee. 2009. *Discovering the Discriminative Views: Measuring Term Weights for Sentiment Analysis.* ACL2009.

Ramanathan Narayanan, Bing Liu and Alok choudhary. 2009. *Sentiment Analysis of Conditional Sentences.* EMNLP-2009.

Bo Pang and Lillian Lee. 2004. *A sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.* ACL-2004.

Bo Pang, Lillian Lee, Vaithyanathan Shivakumar. 2002. *Thumbs up? Sentiment Classification Using Machine Learning Techniques.* EMNLP-2002.

Carlo Strapparava and Alessandro Valitutt. 2004. *WordNet-Affect: an Affective Extension of WordNet.* LREC-2004.

Peter Turney. 2002. *Thumbs up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews.* ACL-2002.

Janyce Wiebe and Rada Mihalcea. 2006. *Word Sense and Subjectivity.* Proceedings of COLING-ACL2006.