# Exploring Feature Definition and Selection for Sentiment Classifiers

**Yelena Mejova**[1] and **Padmini Srinivasan**[1,2]
[1]Computer Science, [2]Management Sciences, University of Iowa, Iowa City, IA
yelena-mejova, padmini-srinivasan@uiowa.edu

**Abstract**

In this paper, we systematically explore feature definition and selection strategies for sentiment polarity classification. We begin by exploring basic questions, such as whether to use stemming, term frequency versus binary weighting, negation-enriched features, n-grams or phrases. We then move onto more complex aspects including feature selection using frequency-based vocabulary trimming, part-of-speech and lexicon selection (three types of lexicons), as well as using expected Mutual Information (MI). Using three product and movie review datasets of various sizes, we show, for example, that some techniques are more beneficial for larger datasets than the smaller. A classifier trained on only few features ranked high by MI outperformed one trained on all features in large datasets, yet in small dataset this did not prove to be true. Finally, we perform a space and computation cost analysis to further understand the merits of various feature types.

## Introduction

Text polarity classification is one of the main tasks for Sentiment Analysis (SA), a field that has seen much growth over the past decade. Much has been written on the usefulness of various feature definition techniques for SA, however, it is still unclear which features are the best. For example, regarding document representation, the literature contains multiple (sometimes conflicting) studies concerning the usefulness of different types of features.

To better understand the merit of current techniques, we study features for sentiment analysis along two dimensions. First, we examine the basic units extracted from texts: words, n-grams, and phrases. Second, we explore feature selection, considering both frequency-based and probabilistic strategies. Here, besides parts of speech (POS) we explore three different lexicons: one extracted from Affect Control Theoretical sociological studies of emotion (Mejova 2010), and two extensions of WordNet: SentiWordNet (Esuli and Sebastiani 2006) and WordNet-Affect (Strapparava and Vlitutti 2004).

We test these techniques on three datasets of various sizes. We show that the size of the dataset affects the performance of some of the techniques. For example, using top few thousand features using mutual Information for large datasets improves the performance, whereas it proves too selective for the smaller dataset. Similarly, the effect of frequency-based feature selection on classifier performance differs for each dataset.

Finally, because a marginal improvement in performance may be overshadowed by the cost of computing the feature, we present cost analysis for each of the features in terms of processing time and storage space.



## Experimental Setup

We perform tests on three datasets. First comes from (Pang and Lee 2004) and includes 1000 positive and 1000 negative movie reviews from IMDB. Second dataset comes from (Jindal and Liu 2007) and is a sample of 20,000 product reviews (taken out of 5,838,855 original documents for tractability). We sampled according to the polarity proportions in the original dataset, taking reviews with rating 5 to be positive (17,480) and 1 to be negative (2,520). The third dataset is a subset of another multi-domain sentiment dataset which has been used in (Blitzer, Dredze, and Pereira 2007) with 21,972 positive and 16,576 negative documents. Note that the last two datasets have unequal number of positive and negative reviews.

Classification was done using Weka sequential minimal optimization (SMO) algorithm for training a support vector classifiers (Platt 1998). We use an SVM for classification for two reasons. First, it is not our intention to determine the best classifier for the task, but the best feature set. Second, SVMs have been widely used in SA and in many cases outperform all other classifiers (Li and Zong 2008; Pang and Lee 2002). Our classifier was tested using 10-fold cross-validation.

## Feature Definition

In this section, we present our results for the different features and discuss their potential usefulness in polarity classification.

Table 1 presents classifier performance scores in terms of overall accuracy, and the F-measure (which combines information about both precision and recall) for negative and positive classes.

**Words versus Stems** Though one may certainly represent a document by the raw words in it, a classic technique in information retrieval is to stem the words to their morphological roots. Stemmed feature vectors are smaller in size, since

Table 1: Performance for single-word and n-gram features

| Run # | Stem-ming | TF vs binary | Neg. words | n-gram | Pang & Lee | | | Jindal | | | Blitzer | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Acc | $F_n$ | $F_p$ | Acc | $F_n$ | $F_p$ | Acc | $F_n$ | $F_p$ |
| 1 | no | TF | no | – | 0.858 | 0.860 | 0.856 | 0.926 | 0.655 | 0.959 | 0.864 | 0.841 | 0.881 |
| 2 | yes | TF | no | – | 0.848 | 0.849 | 0.847 | 0.925 | 0.655 | 0.958 | 0.862 | 0.839 | 0.880 |
| 3 | yes | bin | no | – | 0.841 | 0.841 | 0.841 | 0.926 | 0.684 | 0.958 | 0.858 | 0.835 | 0.875 |
| 4 | no | bin | no | – | 0.859 | 0.859 | 0.858 | 0.925 | 0.677 | 0.958 | 0.859 | 0.836 | 0.876 |
| 5 | no | TF | yes | – | 0.866 | 0.868 | 0.864 | 0.929 | 0.667 | 0.960 | 0.867 | 0.845 | 0.884 |
| 6 | no | TF | no | 2 | 0.851 | 0.858 | 0.843 | 0.910 | 0.496 | 0.951 | 0.855 | 0.825 | 0.877 |
| 7 | no | TF | no | 3 | 0.788 | 0.816 | 0.751 | 0.877 | 0.075 | 0.934 | 0.816 | 0.776 | 0.832 |
| 8 | no | TF | no | 1,2 | **0.875** | **0.879** | **0.869** | 0.913 | 0.547 | 0.952 | 0.879 | 0.856 | 0.896 |
| 9 | no | TF | no | 1,2,3 | 0.830 | 0.843 | 0.815 | **0.947** | **0.748** | **0.970** | **0.896** | **0.876** | **0.910** |
| 10 | no | TF | no | phrase | 0.767 | 0.783 | 0.749 | 0.881 | 0.228 | 0.936 | 0.813 | 0.768 | 0.844 |
| bl | majority rule | | | | 0.500 | 0.500 | 0.500 | 0.779 | 0.126 | 0.874 | 0.510 | 0.430 | 0.570 |

they aggregate across occurrences of variants of a given word. Stemming has had mixed success in both information retrieval and text mining, and as (Dave, Lawrence, and Pennock 2003) we do not find it valuable for the task of polarity classification. By not stemming the terms in run 1, the accuracy improves on average, but insignificantly compared to run 2. Although the improvement is more pronounced for Pang & Lee dataset, with an increase of significance at $p = 0.055$ between runs 3 and 4 (which are otherwise identical).

**Binary versus Term Frequency Weights** A standard approach in information retrieval is to use term frequency (TF) weights to indicate the relative importance of features in document representations. However, some research has shown that binary weighting (0 if the word appears in the document, 1 otherwise) is more beneficial for polarity classification (Pang and Lee 2002). In a study of the standard information retrieval weighting schemes in SA, (Paltoglou and Thelwall 2010) found that using binary features is better than raw term frequency, though a scaled TF version performs as well as binary.

Comparing run 2 (TF) to run 3 (binary weights) as well as run 1 to run 4, we see insignificant changes in performance for all datasets. Note that there is, however a significant change in the F-measure for the negative class in Jindal dataset. Recall that this dataset is the most challenging as it contains only 12.6% negative documents, resulting in a lower classification performance for this under-represented class. Because the minority class is often of interest, features that help classifying it bears study in further research.

**Negations** Negations such as *not* and *never* are often included in stopword lists, and hence are removed from the text analysis. Combined with other words, though, negations reverse the polarity of words. Because polarity classification depends so much on negations, SA researchers have tried incorporating them into the feature vector. We take the approach of (Das and Chen 2001) who use a heuristic to identify negated words and create a new feature by appending *NOT-* to the words (for example, a phrase "don't like" results in feature *NOT-like*). Alas, adding negated-word features in run 5 has proven to be marginally useful. Compared to otherwise identical run 1, the improvement has been made at insignificance levels for all of the three datasets.

**N-grams** Negation phrases discussed above can be considered as a special case of n-grams, which are ordered sets of words. The benefit of using n-grams instead of single words as features comes in being able to capture some dependencies between the words and the importance of individual phrases.

Runs 6 through 9 include n-gram features of $n$ up to 3 (generated using CMU Toolkit http://www.speech.cs.cmu.edu). To test the effect of each level of $n$, all other aspects of the feature space were kept constant. It is clear that the higher $n$-grams alone decrease the accuracy for all datasets. Run 8, which includes 1- and 2-gram features, performs the best for the smallest dataset, and run 9, which includes 1-, 2-, and 3-grams, is best for the other two. These results suggest that the $n$ should be chosen appropriately for the size of the dataset.

**Phrases** Since n-grams are often synthetic, in that they do not necessarily represent a semantically cohesive part of text, we explore the use of grammatical phrases as features. Using a CRF-based phrase chunker (http://jtextpro.sourceforge.net/), we break the text into phrases and use these as features. Like 2- and 3-grams, phrases alone do not outperform run 1. Further study is needed to determine the quality of the phrases produced by the tool, and possible benefits of using this feature space in combination with others.

## Feature Selection

**Frequency-Based Selection** In text modeling, it is often the practice to remove words which appear rarely in the corpus. These are presumed to be perhaps misspellings that do not help in generalization during classification. On the other hand, words that occur only once in a given corpus have been found to be high-precision indicators of subjectivity (Wiebe et al. 2004).

In Figure 1 we explore the merits of cutting off the "tail" of the vocabulary, that is, excluding the terms that appear fewer than $c$ times in the dataset from the feature space. The decrease in the performance compared to full-vocabulary

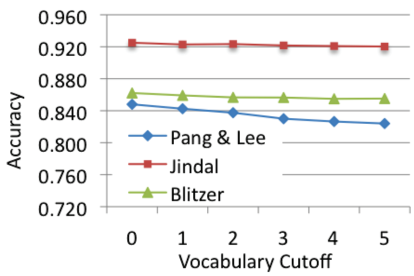
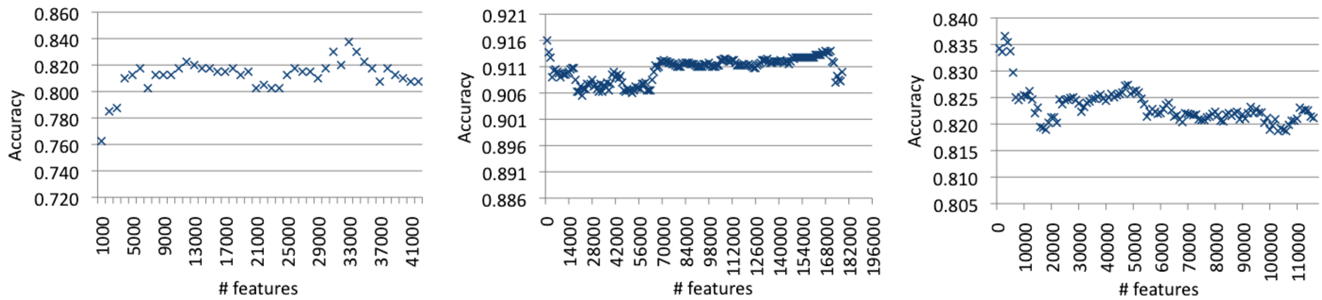



Figure 1: Feature selection using frequency-based vocabulary cut-offs

Table 2: POS and Lexicon-based feature selection for single-word features

| | Pang & Lee | | Jindal | | Blitzer | |
|---|---|---|---|---|---|---|
| **Run** | **Acc** | **# features** | **Acc** | **# features** | **Acc** | **# features** |
| ADJ | 0.781 | 13,546 | 0.901 | 21,150 | 0.772 | 16,217 |
| VB | 0.690 | 11,845 | 0.885 | 20,739 | 0.748 | 16,853 |
| NN | 0.756 | 26,965 | 0.882 | 84,510 | 0.758 | 60,034 |
| ADJ ∪ VB ∪ NN | 0.846 | 43,223 | 0.921 | 111,675 | 0.851 | 81,095 |
| ACT | 0.678 | 3997 | 0.902 | 3997 | 0.674 | 3997 |
| SWN | 0.819 | 52902 | 0.875 | 52902 | 0.797 | 52902 |
| WNA | 0.693 | 2367 | 0.876 | 2367 | 0.656 | 2367 |
| run 1 | 0.858 | 50,917 | 0.926 | 218,103 | 0.864 | 153,789 |
| majority | 0.500 | — | 0.779 | — | 0.510 | — |




(a) Pang & Lee (b) Jindal (c) Blitzer

Figure 2: Performance with MI feature selection at various cut-offs

run was not significant at $p < 0.05$ level up to $c = 3$ for Pang & Lee, $c = 4$ for Jindal, and $c = 1$ for Blitzer datasets (that is, when words appearing $c$ times or less were excluded). This means that we can get an equivalent performance from a classifier for Jindal dataset while excluding words that appear 4 times or less in the dataset (leaving only 15.3% of original vector set!). Notice the differing acceptable cutoffs for the three datasets, which suggests that classification of some datasets is more sensitive to rare words than of others.

**Mutual Information Based Selection** The performance of the classifier may also be improved by removing some of the less useful features. We use expected Mutual Information as a measurement of a feature's usefulness. We divide each dataset into training (60%), tuning (20%), and testing (20%) subsets. Features were extracted from the training set and ordered by their MI scores. Top $N$ were chosen to represent the documents in the tuning set, with $N$ varying from top few features to the size of the feature space. Finally, for each dataset an $N$ was chosen to maximize performance, and the testing set was used to determine classifier performance at this cutoff.

Figure 2 shows the performance of the classifiers at various cutoff points for the tuning sets. For all datasets, the performance drops off as the number of features approaches 100% (the number of features in full feature space is different for each dataset). This means that when sorted by MI, the bottom features hurt the performance of the classifier. Towards the top of the list, the performance differs between the relatively small Pang & Lee dataset and the others, which are larger by an order of magnitude. We noted the best cutoff point for each dataset and use the testing set to get the accuracy scores of 0.798 (Pang & Lee) at 76% cutoff, 0.911 (Jindal) at 1%, and 0.837 (Blitzer) at 3%.

**Part of Speech-Based Selection** In particular for SA, certain POS have been determined to be more useful in classification tasks. For example, adjectives, adverbs, and verbs have been used for sentiment classification (Benamara et al. 2007; Chesley et al. 2006). If indeed adjectives are important factors in predicting sentiment polarity, limiting the feature space to only these may improve classifier performance by removing less useful words. We test this notion by retaining only words that are adjectives, verbs, and nouns individually and in combination. Results can be seen in Table 2. For each dataset besides accuracy we present the number of features for each run. Although the best accuracy is achieved when all three parts of speech are used, the best improvement attained per feature is with adjectives, and secondly with verbs, showing that these two parts of speech are indeed more helpful in polarity classification.

**Lexicon-Based Selection** Similarly, sentiment-annotated lexicons may be used for feature selection. By selecting terms which are indicative of strong sentiment, less useful features may be excluded from the feature set. Popular lexicons are the extensions of WordNet (http://wordnet.princeton.edu/), a large lexical database of English. SentiWordNet, for example, contains polarity and objectivity labels for the WordNet terms (Esuli and Sebas-

Table 3: Space and computation time statistics for various features for Pang & Lee dataset

| | Space to store... | | Time to generate... (ms) | |
|---|---|---|---|---|
| **Feature type** | **# of features** | **space (bytes)** | **feature space** | **doc vector** |
| Single-word | 50,918 | 6,513,249 | 5,917 | 584 |
| Negation-enriched | 2,305 | 143,923 | 7,519 | 244 |
| 2-grams | 468,023 | 24,142,950 | 7,483 | 4,254 |
| 3-grams | 1,044,171 | 41,152,199 | 11,245 | 8,625 |
| Phrases | 171,515 | 8,026,851 | 141,012 | 1,151 |

tiani 2006). In WordNet-Affect (Strapparava and Vlitutti 2004) take advantage of synsets - word groupings in WordNet - to label each synset with affective labels. Both have been widely used in the community, and we use both lexicons in our analysis. Furthermore, we use a lexicon derived from sociological studies on emotion, which we call the ACT (Affect Control Theory) lexicon (Mejova 2010). The Affect Control Theory (ACT), SentiWordNet (SWN) and WordNet-Affect (WNA) lexicons contain 3997, 52902, and 2367 terms, respectively. The largest lexicon, SWN, provides the best performance for Pang & Lee and Blitzer datasets. Yet in Jindal its performance is equivalent to the WNA run, making its improvement/feature ratio 25 times less than that of the WNA run.

## Cost Analysis

Finally, we analyze the computation time needed to generate the various features and the space needed to store them. The first two columns of Table 3 show the number of features and size of the standard Weka ARFF file containing them (in sparse format) for Pang & Lee dataset. The largest files produced by far were the n-grams, followed by phrases. The last two columns show the time (in milliseconds) it takes to generate the feature space and the average time it takes to generate a feature vector for each document. The tests were run on a computer with AMD Athlon 64 Processor with 1024KB cache and 1GB RAM. Although in terms of number of features negation-enriched features are few compared to the other types of features, because templates are used to extract these, the time it takes to generate the feature space is even greater than that of generating the 2-gram feature space.

## Conclusion

In our exploration of some of the latest popular feature definition and selection techniques, we use three datasets to test techniques popular in SA literature. We confirm some hypotheses, including that adjectives are important for polarity classification, and that stemming and using binary instead of term frequency feature vectors do not impact performance. We also show that the helpfulness of certain techniques depends on the nature of the dataset, including its size and class balance. Finally, we present the cost analysis in terms of space used to store the dataset and the time it takes to compute it. We see that, for example, it takes more time to compute negation-enriched features (using templates) than it takes to compute the whole vocabulary, putting in question any benefit these may give when working with large datasets.

## References


Benamara, F.; Cesarano, C.; Picariello, A.; Reforgiato, D.; and Subrahmanian, V. 2007. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. *Proc. of ICWSM*.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *ACL* 440–447.

Chesley, P.; Vincent, B.; Xu, L.; and Srihari, R. K. 2006. Using verbs and adjectives to automatically classify blog sentiment. *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*.

Das, S., and Chen, M. 2001. Yahoo! for amazon: Extracting market sentiment from stock message boards. *Proc. of APFA*.

Dave, K.; Lawrence, S.; and Pennock, D. M. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proc. of WWW*.

Esuli, A., and Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. *Proc. of LREC*.

Jindal, N., and Liu, B. 2007. Review span detection. *WWW*.

Li, S., and Zong, C. 2008. Multi-domain sentiment classification. *HLT-Short '08 Proc. of ACL on Human Language Technologies*.

Mejova, Y. 2010. Tapping into sociological lexicons for sentiment polarity classification. *Young Scientists Conference, RuSSIR'10*.

Paltoglou, G., and Thelwall, M. 2010. A study of information retrieval weighting schemes for sentiment analysis. *Proc. of ACL* 1386–1395.

Pang, B., and Lee, L. 2002. Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing* 10:79–86.

Pang, B., and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *Proc. of ACL*.

Platt, J. C. 1998. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods - Support Vector Learning*.

Strapparava, C., and Vlitutti, A. 2004. Wordnet-affect: and affective extension of wordnet. *Proc. of LREC*.

Wiebe, J. M.; Wilson, T.; Bruce, R.; Bell, M.; and Martin, M. 2004. Learning subjective language. *Computational Linguistics* 30.