# Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms

**Michael Gamon**
Natural Language Processing Group
Microsoft Research
mgamon@microsoft.com

**Anthony Aue**
Natural Language Processing Group
Microsoft Research
anthaue@microsoft.com

## Abstract

We describe an extension to the technique for the automatic identification and labeling of sentiment terms described in Turney (2002) and Turney and Littman (2002). Their basic assumption is that sentiment terms of similar orientation tend to co-occur at the document level. We add a second assumption, namely that sentiment terms of opposite orientation tend **not** to co-occur at the sentence level. This additional assumption allows us to identify sentiment-bearing terms very reliably. We then use these newly identified terms in various scenarios for the sentiment classification of sentences. We show that our approach outperforms Turney's original approach. Combining our approach with a Naive Bayes bootstrapping method yields a further small improvement of classifier performance. We finally compare our results to precision and recall figures that can be obtained on the same data set with labeled data.

## 1 Introduction

The field of sentiment classification has received considerable attention from researchers in recent years (Pang and Lee 2002, Pang et al. 2004, Turney 2002, Turney and Littman 2002, Wiebe et al. 2001, Bai et al. 2004, Yu and Hatzivassiloglou 2003 and many others). The identification and classification of sentiment constitutes a problem that is orthogonal to the usual task of text classification. Whereas in traditional text classification the focus is on topic identification, in sentiment classification the focus is on the assessment of the writer's sentiment toward the topic.

Movie and product reviews have been the main focus of many of the recent studies in this area (Pang and Lee 2002, Pang et al. 2004, Turney 2002, Turney and Littman 2002). Typically, these reviews are classified at the document level, and the class labels are "positive" and "negative". In this work, in contrast, we narrow the scope of investigation to the sentence level and expand the set of labels, making a threefold distinction between "positive", "neutral", and "negative". The narrowing of scope is motivated by the fact that for realistic text mining on customer feedback, the document level is too coarse, as described in Gamon et al. (2005). The expansion of the label set is also motivated by real-world concerns; while it is a given that review text expresses positive or negative sentiment, in many cases it is necessary to also identify the cases that don't carry strong expressions of sentiment at all.

Traditional approaches to text classification require large amounts of labeled training data. Acquisition of such data can be costly and time-consuming. Due to the highly domain-specific nature of the sentiment classification task, moving from one domain to another typically requires the acquisition of a new set of training data. For this reason, unsupervised or very weakly supervised methods for sentiment classification are especially



*Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, pages 57–64,
Ann Arbor, June 2005. 

desirable.[1] Our focus, therefore, is on methods that require very little data annotation.

We describe a method to automatically identify the sentiment vocabulary in a domain. This method rests on three special properties of the sentiment domain:

1. the presence of certain words can serve as a proxy for the class label
2. sentiment terms of *similar* orientation tend to co-occur
3. sentiment terms of *opposite* orientation tend to not co-occur at the sentence level.

Turney (2002) and Turney and Littman (2002) exploit the first two generalizations for unsupervised sentiment classification of movie reviews. They use the two terms *excellent* and *poor* as seed terms to determine the semantic orientation of other terms. These seed terms can be viewed as proxies for the class labels "positive" and "negative", allowing for the exploitation of otherwise unlabeled data: Terms that tend to co-occur with *excellent* in documents tend to be of positive orientation, and vice versa for *poor*. Turney (2002) starts from a small (2 word) set of terms with known orientation (*excellent* and *poor*). Given a set of terms with unknown sentiment orientation, Turney (2002) then uses the PMI-IR algorithm (Turney 2001) to issue queries to the web and determine, for each of these terms, its pointwise mutual information (PMI) with the two seed words across a large set of documents. Term candidates are constrained to be adjectives, which tend to be the strongest bearers of sentiment. The sentiment orientation (SO) of a term is then determined by the difference between its association (PMI) with the positive seed term *excellent* and its association with the negative seed term *poor*. The resulting list of terms and associated sentiment orientations can then be used to implement a classifier: semantic orientation of the terms in a document of unknown sentiment is added up, and if the overall score is positive, the document is classified as being of positive sentiment, otherwise it is classified as negative.

Yu and Hatzivassiloglou (2003) extend this approach by (1) applying it at the sentence level (instead of the document-level), (2) taking into account non-adjectival parts-of-speech, and (3) using larger sets of seed words. Their classification goal also differs from Turney's: it is to distinguish opinion sentences from factual statements.

Turney et al.'s approach is based on the assumption that sentiment terms of similar orientation tend to co-occur in documents. Our approach takes advantage of a second assumption: At the sentence level, sentiment terms of opposite orientation tend *not* to co-occur. This is, of course, an assumption that will only hold in general, with exceptions. Basically, the assumption is that sentences of the following form:

*I dislike X.*
*I really like X.*

are more frequent than "mixed sentiment" sentences such as

*I dislike X but I really like Y.*

It has been our experience that this generalization does hold often enough to be useful.

We propose to utilize this assumption to identify a set of sentiment terms in a domain. We select the terms that have the lowest PMI scores on the sentence level with respect to a set of manually selected seed words. If our assumption about low association at the sentence level is correct, this set of low-scoring terms will be particularly rich in sentiment terms. We can then use this newly identified set to:

(1) use Turney's method to find the orientation for the terms and employ the terms and their scores in a classifier, and
(2) use Turney's method to find the orientation for the terms and add the new terms as additional seed terms for a second iteration

As opposed to Turney (2002), we do not use the web as a resource to find associations, rather we apply the method directly to in-domain data. This has the disadvantage of not being able to apply the classification to any arbitrary domain. It is worth noting, however, that even in Turney (2002) the choice of seed words is explicitly motivated by domain properties of movie reviews.

In the remainder of the paper we will describe results from various experiments based on this assumption. We also show how we can combine this method with a Naive Bayes bootstrapping approach that takes further advantage of the unlabeled data (Nigam et al. 2000).

[1] For domain-specificity of sentiment classification see Engström (2004) and Aue and Gamon (2005).

## 2 Data

For our experiments we used a set of car reviews from the MSN Autos web site. The data consist of 406,818 customer car reviews written over a four-year period. Aside from filtering out examples containing profanity, the data was not edited. The reviews range in length from a single sentence (56% of all cases) to 50 sentences (a single review). Less than 1% of reviews contain ten or more sentences. There are almost 900,000 sentences in total. When customers submitted reviews to the website, they were asked for a recommendation on a scale of 1 (negative) to 10 (positive). The average score was very high, at 8.3, yielding a strong skew in favor of positive class labels. We annotated a randomly-selected sample of 3,000 sentences for sentiment. Each sentence was viewed in isolation and classified as positive, negative or neutral. The neutral category was applied to sentences with no discernible sentiment, as well as to sentences that expressed both positive and negative sentiment. Three annotators had pair-wise agreement scores (Cohen's Kappa score, Cohen 1960) of 70.10%, 71.78% and 79.93%, suggesting that the task of sentiment classification on the sentence level is feasible but difficult even for people. This set of data was split into a development test set of 400 sentences and a blind test set of 2600 sentences.

Sentences are represented as vectors of binary unigram features. The total number of observed unigram features is 72988. In order to restrict the number of features to a manageable size, we disregard features that occur less than 10 times in the corpus. With this restriction we obtain a reduced feature set of 13317 features.

## 3 Experimental Setup

Our experiments were performed as follows: We started with a small set of manually-selected and annotated seed terms. We used 4 positive and 6 negative seed terms. We decided to use a few more negative seed words because of the inherent positive skew in the data that makes the identification of negative sentences particularly hard. The terms we used are:

| **positive:** | **negative:** |
|---|---|
| good | bad |
| excellent | lousy |
| love | terrible |
| happy | hate |
| | suck |
| | unreliable |

There was no tuning of the set of initial seed terms; the 10 words were originally chosen intuitively, as words that we observed frequently when manually inspecting the data.

We then used these seed terms in two basic ways: (1) We used them as seeds for a Turney-style determination of the semantic orientation of words in the corpus (semantic orientation, or SO method). As mentioned above, this process is based on the assumption that terms of similar orientation tend to co-occur. (2) We used them to mine sentiment vocabulary from the unlabeled data using the additional assumption that sentiment terms of opposite orientation tend not to co-occur at the sentence level (sentiment mining, or SM method). This method yields a set of sentiment terms, but no orientation for that set of terms. We continue by using the SO method to find the semantic orientation for this set of sentiment terms, effectively using SM as a feature selection method for sentiment terminology.

Pseudo-code for the SO and SM approaches is provided in Figure 1 and Figure 2. As a first step for both SO and SM methods (not shown in the pseudocode), PMI needs to be calculated for each pair (f, s) of feature *f* and seed word *s* over the collection of feature vectors.

**Semantic Orientation (SO) method to find semantic orientation for a set of features F, given a set of feature vectors V:**

```
FOREACH feature f in F:
    FOREACH positive seed word spos
        PosScore(f) = PosScore(f)+PMI(f, spos)
    FOREACH negative seed word sneg
        NegScore(f)+PMI(f, sneg)
    normalize scores by number of positive/negative
    seed features:
        PosScore(f) = PosScore(f)/number of
        positive seed features
        NegScore(f) = NegScore(f)/number of
        negative seed features
    calculate overall semantic orientation (SO) for f:
        SO = PosScore(f)-NegScore(f)
```

Figure 1: SO method for determining semantic orientation

**Mining for sentiment terms (SM) in a set of features F:**

```
FOREACH feature f in F:
        FOREACH seed word s
                IF PMI(f, s) < Score(f):
                        Score(f) = PMI(f, s)

Select the top n% of features f that have the lowest Score(f)
```

Figure 2: SM method for mining sentiment terms

In the first scenario (using straightforward SO), features *F* range over all observed features in the data (modulo the aforementioned count cutoff of 10). In the second scenario (SM + SO), features *F* range over the *n*% of features with the lowest PMI scores with respect to any of the seed words that were identified using the sentiment mining technique in Figure 2.

The result of both SO and SM+SO is a list of unigram features which have an associated semantic orientation score, indicating their sentiment orientation: the higher the score, the more "positive" a term, and vice versa.

This list of features and associated scores can be used to construct a simple classifier: for each sentence with unknown sentiment, we take the sum of the semantic orientation scores for all of the unigrams in that sentence. This overall score determines the classification of the sentence as "positive", "neutral" or "negative" as shown in Figure 3.

**Scoring and classifying sentence vectors:**

```
(1) assigning a sentence score:
FOREACH feature f in sentence vector v:
        Score(v) = Score(v) + SO(f)

(2) assigning a class label based on the sentence score:
IF Score(v) > threshold1:
        Class(v) = "positive"
ELSE IF Score(v) < threshold1 AND Score(v) > threshold2:
        Class(v) = "neutral"
ELSE
        Class(v) = "negative"
```

Figure 3: Using SO scores for sentence scoring and classification

The two thresholds used in classification need to be determined empirically by taking the distribution of class values in the corpus into account. For our experiments we simply took the distribution of class labels in the 400 sentence development test set as an approximation of the overall class label distribution: we determined that distribution to be 15.5% for negative sentences, 21.5% for neutral sentences, and 63.0% for positive sentences. Scores for all sentence vectors in the corpus are then collected using the scoring part of the algorithm in Figure 3. The scores are sorted and the thresholds are determined as the cutoffs for the top 63% and bottom 15.5% of scores respectively.

## 4 Results

### 4.1. Comparing SO and SM+SO

In our first set of experiments we manipulated the following parameters:

1. the choice of SO or SM+SO method
2. the choice of *n* when selecting the *n*% semantic terms with lowest PMI score in the SM method

The tables below show the results of classifying sentence vectors using the unigram features and associated scores produced by SO and SO+SM. We used the 2,600-sentence manually-annotated test set described previously to establish these numbers. Since the data exhibit a strong skew in favor of the positive class label, we measure performance not in terms of accuracy but in terms of average precision and recall across the three class labels, as suggested in (Manning and Schütze 2002).

| | Avg precision | Avg recall |
|---|---|---|
| SO | 0.4481 | 0.4511 |

Table 1: Using the SO approach.

Table 1 shows results of using the SO method on the data. Table 2 presents the results of combining the SM and SO methods for different values of *n*. The best results are shown in boldface.

As a comparison between Table 1 and Table 2 shows, the highest average precision and recall scores were obtained by combining the SM and SO methods. Using SM as a feature selection mechanism also reduces the number of features significantly. While the SO method employed on sentence-level vectors uses 13,000 features, the best-performing SM+SO combination uses only 20% of this feature set, indicating that SM is indeed effective in selecting the most important sentiment-bearing terms.

We also determined that the positive impact of SM is not just a matter of reducing the number of features. If SO - without the SM feature selection step - is reduced to a comparable number of features by taking the top features according to absolute score, average precision is at 0.4445 and average recall at 0.4464.

| | N=10 | | N=20 | | N=30 | | N=40 | | N=50 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Avg prec | Avg rec | Avg prec | Avg rec | Avg prec | Avg rec | Avg prec | Avg rec | Avg prec | Avg rec |
| SM+SO SO from document level | 0.4351 | 0.4377 | **0.4568** | **0.4605** | 0.4528 | 0.4557 | 0.4457 | 0.4478 | 0.4451 | 0.4475 |

Table 2: combining SM and SO.

| **Sentiment terms in top 100 SM terms** | **Sentiment terms in top 100 SO terms** |
|---|---|
| excellent, terrible, broke, junk, alright, bargain, grin, highest, exceptional, exceeded, horrible, loved, waste, ok, death, leaking, outstanding, cracked, rebate, warped, hooked, sorry, refuses, excellant, satisfying, died, biggest, competitive, delight, avoid, awful, garbage, loud, okay, competent, upscale, dated, mistake, sucks, superior, high, kill, neither | excellent, happy, stylish, sporty, smooth, love, quiet, overall, pleased, plenty, dependable, solid, roomy, safe, good, easy, smaller, luxury, comfortable, style, loaded, space, classy, handling, joy, small, comfort, size, perfect, performance, room, choice, recommended, package, compliments, awesome, unique, fun, holds, comfortably, extremely, value, free, satisfied, little, recommend, limited, great, pleasure |
| **Non sentiment terms in top 100 SM terms** | **Non sentiment terms in top 100 SO terms** |
| alternative, wont, below, surprisingly, maintained, choosing, comparing, legal, vibration, seemed, claim, demands, assistance, knew, engineering, accelleration, ended, salesperson, performed, started, midsize, site, gonna, lets, plugs, industry, alternator, month, told, vette, 180, powertrain, write, mos, walk, causing, lift, es, segment, $250, 300m, wanna, february, mod, $50, nhtsa, suburbans, manufactured, tiburon, $10, f150, 5000, posted, tt, him, saw, jan, | condition, very, handles, milage, definitely, definately, far, drives, shape, color, price, provides, options, driving, rides, sports, heated, ride, sport, forward, expected, fairly, anyone, test, fits, storage, range, family, sedan, trunk, young, weve, black, college, suv, midsize, coupe, 30, shopping, kids, player, saturn, bose, truck, town, am, leather, stereo, car, husband |

Table 3: the top 100 terms identified by SM and SO

Table 3 shows the top 100 terms that were identified by each SM and SO methods. The terms are categorized into sentiment-bearing and non-sentiment bearing terms by human judgment. The two sets seem to differ in both strength and orientation of the identified terms. The SM-identified words have a higher density of negative terms (22 out of 43 versus 2 out of 49 for the SO-identified terms). The SM-identified terms also express sentiment more strongly, but this conclusion is more tentative since it may be a consequence of the higher density of negative terms.

## 4.2. Multiple iterations: increasing the number of seed features by SM+SO

In a second set of experiments, we assessed the question of whether it is possible to use multiple iterations of the SM+SO method to gradually build the list of seed words. We do this by adding the top n% of features selected by SM, along with their orientation as determined by SO, to the initial set of seed words. The procedure for this round of experiments is as follows:

- take the top n% of features identified by SM (we used n=1 for the reported re-

sults, since preliminary experiments with other values for n did not improve results)

- perform SO for these features to determine their orientation
- take the top 15.5% negative and top 63% positive (according to class label distribution in the development test set) of the features and add them as negative/positive seed features respectively

This iteration increases the number of seed features from the original 10 manually-selected features to a total of 111 seed features.

With this enhanced set of seed features we then re-ran a subset of the experiments in Table 2. Results are shown in Table 4. Increasing the number of seed features through the SM feature selection method increases precision and recall by several percentage points. In particular, precision and recall for negative sentences are boosted.

| | Avg precision | Avg recall |
|---|---|---|
| SM + SO, n=10, SO from document vectors | 0.4826 | 0.48.76 |
| SM + SO, n=30, SO from document vectors | **0.4957** | **0.4995** |
| SM + SO, n=50, SO from document vectors | 0.4914 | 0.4952 |

Table 4: Using 2 iterations to increase the seed feature set

We also confirmed that these results are truly attributable to the use of the SM method for the first iteration. If we take an equivalent number of features with strongest semantic orientation according to the SO method and add them to the list of seed features, our results degrade significantly (the resulting classifier performance is significantly different at the 99.9% level as established by the McNemar test). This is further evidence that SM is indeed an effective method for selecting sentiment terms.

### 4.3. Using the SO classifier to bootstrap a Naive Bayes classifier

In a third set of experiments, we tried to improve on the results of the SO classifier by combining it with the bootstrapping approach described in (Nigam et al. 2000). The basic idea here is to use the SO classifier to label a subset of the data $D_L$. This labeled subset of the data is then used to bootstrap a Naive Bayes (NB) classifier on the remaining unlabeled data $D_U$ using the Expectation Maximization (EM) algorithm:

(1) An initial naive Bayes classifier with parameters $\theta$ is trained on the documents in $D_L$.
(2) This initial classifier is used to estimate a probability distribution over all classes for each of the documents in $D_U$. (E-Step)
(3) The labeled and unlabeled data are then used to estimate parameters for a new classifier. (M-Step)

Steps 2 and 3 are repeated until convergence is achieved when the difference in the joint probability of the data and the parameters falls below the configurable threshold $\varepsilon$ between iterations. Another free parameter, $\lambda$, can be used to control how much weight is given to the unlabeled data.

For our experiments we used classifiers from the best SM+SO combination (2 iterations at n=30) from Table 4 above to label 30% of the total data. Table 5 shows the average precision and recall numbers for the converged NB classifier.[2] In addition to improving average precision and recall, the resulting classifier also has the advantage of producing class probabilities instead of simple scores.[3]

| | Avg precision | Avg recall |
|---|---|---|
| Bootstrapped NB classifier | 0.5167 | 0.52 |

Table 5: Results obtained by bootstrapping a NB classifier

### 4.4. Results from supervised learning: using small sets of labeled data

Given infinite resources, we can always annotate enough data to train a classifier using a supervised algorithm that will outperform unsupervised or weakly-supervised methods. Which approach to take depends entirely on how much time and money are available and on the accuracy requirements for the task at hand.

[2] In this experiment, $\lambda$ was set to 0.1 and $\varepsilon$ was set to 0.05.
[3] We also experimented with labeling the whole data set with the best of our SO score classifiers, and then training a linear Support Vector Machine classifier on the data. The results were considerably worse than any of the reported numbers, so they are not included in this paper.

To help situate the precision and recall numbers presented in the tables above, we trained Support Vector Machines (SVMs) using small amounts of labeled data. SVMs were trained with 500, 1000, 2000, and 2500 labeled sentences. Annotating 2500 sentences represents approximately eight person-hours of work. The results can be found in Table 5. We were pleasantly surprised at how well the unsupervised classifiers described above perform in comparison to state-of-the-art supervised methods (albeit trained on small amounts of data).

| Labeled examples | Avg. Precision | Avg. Recall |
|---|---|---|
| 500 | .4878 | .4967 |
| 1000 | .5161 | .5105 |
| 2000 | .5297 | .5256 |
| 2500 | .5017 | .5083 |

Table 6: Average precision and recall for SVMs for small numbers of labeled examples

### 4.5. Results on the movie domain

We also performed a small set of experiments on the movie domain using Pang and Lee's 2004 data set. This set consists of 2000 reviews, 1000 each of very positive and very negative reviews. Since this data set is balanced and the task is only a two-way classification between positive and negative reviews, we only report accuracy numbers here.

| | accuracy | Training data |
|---|---|---|
| Turney (2002) | 66% | unsupervised |
| Pang & Lee (2004) | 87.15% | supervised |
| Aue & Gamon (2005) | 91.4% | supervised |
| SO | 73.95% | unsupervised |
| SM+SO to increase seed words, then SO | 74.85% | weakly supervised |

Table 7: Classification accuracy on the movie review domain

Turney (2002) achieves 66% accuracy on the movie review domain using the PMI-IR algorithm to gather association scores from the web. Pang and Lee (2004) report 87.15% accuracy using a unigram-based SVM classifier combined with subjectivity detection. Aue and Gamon (2005) use a simple linear SVM classifier based on unigrams, combined with LLR-based feature reduction, to achieve 91.4% accuracy. Using the Turney SO method on in-domain data instead of web data achieves 73.95% accuracy (using the same two seed words that Turney does). Using one iteration of SM+SO to increase the number of seed words, followed by finding SO scores for all words with respect to the enhanced seed word set, yields a slightly higher accuracy of 74.85%. With additional parameter tuning, this number can be pushed to 76.4%, at which point we achieve statistical significance at the 0.95 level according to the McNemar test, indicating that there is more room here for improvement. Any reduction of the number of overall features in this domain leads to decreased accuracy, contrary to what we observed in the car review domain. We attribute this observation to the smaller data set.

## 5 Discussion

### 5.1 A note on statistical significance

We used the McNemar test to assess whether two classifiers are performing significantly differently. This test establishes whether the accuracy of two classifiers differs significantly - it does not guarantee significance for precision and recall differences. For the latter, other tests have been proposed (e.g. Chinchor 1995), but time constraints prohibited us from implementing any of those more computationally costly tests.

For the results presented in the previous sections the McNemar test established statistical significance at the 0.99 level over baseline (i.e. the SO results in Table 1) for the multiple iterations results (Table 4) and the bootstrapping approach (Table 5), but not for the SM+SO approach (Table 2).

### 5.2 Future work

This exploratory set of experiments indicates a number of interesting directions for future work. A shortcoming of the present work is the manual tuning of cutoff parameters. This problem could be alleviated in at least two possible ways:

First, using a general combination of the ranking of terms according to SM and SO. In other words, calculate the semantic weight of a term as a combination of SO and its rank in the SM scores.

Secondly, following a suggestion by an anonymous reviewer, the Naive Bayes bootstrapping approach could be used in a feedback loop to inform the SO score estimation in the absence of a manually annotated parameter tuning set.

### 5.3 Summary

Our results demonstrate that the SM method can serve as a valid tool to mine sentiment-rich vocabulary in a domain. SM will yield a list of terms that are likely to have a strong sentiment orientation. SO can then be used to find the polarity for the selected features by association with the sentiment terms of known polarity in the seed word list. Performing this process iteratively by first enhancing the set of seed words through SM+SO yields the best results. While this approach does not compare to the results that can be achieved by supervised learning with large amounts of labeled data, it does improve on results obtained by using SO alone.

We believe that this result is relevant in two respects. First, by improving average precision and recall on the classification task, we move closer to the goal of unsupervised sentiment classification. This is a very important goal in itself given the need for "out of the box" sentiment techniques in business intelligence and the notorious difficulty of rapidly adapting to a new domain (Engström 2004, Aue and Gamon 2005). Second, the exploratory results reported here may indicate a general source of information for feature selection in natural language tasks: features that have a tendency to be in complementary distribution (especially in smaller linguistic units such as sentences) may often form a class that shares certain properties. In other words, it is not only the strong association scores that should be exploited but also the particularly weak (negative) associations.

## References


Anthony Aue and Michael Gamon (2005): "Customizing Sentiment Classifiers to a New Domain: A Case Study. Under review.

Xue Bai, Rema Padman, and Edoardo 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 (MSW 2004), pp 24-35.

Nancy A. Chinchor (1995): Statistical significance of MUC-6 results. Proceedings of the Sixth Message Understanding Conference, pp. 39-44.

J. Cohen (1960): "A coefficient of agreement for nominal scales." In: Educational and Psychological measurements 20, pp. 37–46

Charlotta Engström. 2004. *Topic dependence in Sentiment Classification.* MPhil thesis, University of Cambridge.

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. (2005): "Pulse: Mining Customer Opinions from Free Text". Under review.

Christopher D. Manning and Hinrich Schütze (2002): Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, London.

Kamal Nigam, Andrew McCallum, Sebastian Thrun and Tom Mitchell (2000): Text Classification from Labeled and Unlabeled Documents using EM. In: Machine Learning 39 (2/3), pp. 103-134.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan (2002): "Thumbs up? Sentiment Classification using Machine Learning Techniques". Proceedings of EMNLP 2002, pp. 79-86.

Bo Pang and Lillian Lee. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of ACL 2004, pp.217-278.

Peter D. Turney (2001): "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL." In Proceedings of the Twelfth European Conference on Machine Learning, pp. 491-502.

Peter D. Turney (2002): "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews". In: Proceedings of ACL 2002, pp. 417-424.

Peter D. Turney and M. L. Littman (2002): "Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus." Technical report ERC-1094 (NRC 44929), National Research Council of Canada.

Janyce Wiebe, Theresa Wilson and Matthew Bell (2001): "Identifying Collocations for Recognizing Opinions". In: Proceedings of the ACL/EACL Workshop on Collocation.

Hong Yu and Vasileios Hatzivassiloglou (2003): "Towards Answering opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences". In: Proceedings of EMNLP 2003.