Feature Selection for Sentiment Analysis Based on Content and Syntax Models

Adnan Duric and Fei Song
School of Computer Science, University of Guelph, 50 Stone Road East,
Guelph, Ontario, N1G 2W1, Canada
{aduric,fsong}@uoguelph.ca

Abstract

Recent solutions for sentiment analysis have relied on feature selection methods ranging from lexicon-based approaches where the set of features are generated by humans, to approaches that use general statistical measures where features are selected solely on empirical evidence. The advantage of statistical approaches is that they are fully automatic, however, they often fail to separate features that carry sentiment from those that do not. In this paper we propose a set of new feature selection schemes that use a Content and Syntax model to automatically learn a set of features in a review document by separating the entities that are being reviewed from the subjective expressions that describe those entities in terms of polarities. By focusing only on the subjective expressions and ignoring the entities, we can choose more salient features for document-level sentiment analysis. The results obtained from using these features in a maximum entropy classifier are competitive with the state-of-the-art machine learning approaches.

1 Introduction

As user generated data become more commonplace, we seek to find better approaches to extract and classify relevant content automatically. This gives users a richer, more informative, and more appropriate set of information in an efficient and organized manner. One way for organizing such data is *text classification*, which involves mapping documents into *topical* categories based on the occurrences of particular features. Sentiment Analysis (SA) can be framed as a text classification task where the categories are *polarities* such as *positive* and *negative*. However, the similarities end here. Whereas general text classification is concerned with features that distinguish different topics, sentiment analysis deals with features about subjectivity, affect, emotion, and points-of-view that *describe* or *modify* the related entities. Since user-generated review documents contain both kinds of features, SA solutions ultimately face the challenge of separating the factual content from the subjective content describing it.

For example, taking a segment from a randomly chosen document in Pang et al.'s movie review corpus[1], we see how entities and modifiers are related to each other:

> ... Of course, it helps that **Kaye** has an **actor** as *talented* as **Norton** to play **this part**. It's *astonishing* how *frightening* **Norton** looks with a shaved head and a swastika on his chest. ... Visually, **the film** is *very powerful*. **Kaye** indulges in a lot of *interesting* **artistic choices**, and most of **them** *work nicely*.

Indeed, most of the information about an entity that relates it to a particular polarity comes from the *modifying* words. In the example above, these words are adjectives such as *talented*, *frightening*, *interesting*, and *powerful*. They can also be verbs such as *work* and adverbs such as *nicely*. The entities are

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/

Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT 2011, pages 96–103,
24 June, 2011, Portland, Oregon, USA

represented by various nouns and pronouns such as: *Kaye*, *Norton*, *actor* and *them*.

Therefore, the task of classifying a review document can be explored by taking into account a mixture of entities and their modifiers. An important characteristic of review documents is that the reviewers tend to discuss the whole set of entities throughout the entire document, whereas the modifiers for those entities tend to be more localized at the sentence or phrase level. In other words, each entity can be *polymorphous* within the document, with a long-range *semantic* relationship between its forms while the modifiers in each case are bound to the entity in a short-range, *syntactic* relationship. Generalizing a single entity to all the entities that are found in a document, and taking all their respective modifiers into account, we can start to infer the polarity of the entire document based on the set of all the modifiers. This reduces to finding all the syntactic words in the document and disregarding the entities.

Taking another look at the example modifiers, we might assume that all of the relevant indicators for SA come from specific parts of speech categories such as *adjectives* and *adverbs*, while other parts of speech classes such as nouns are more relevant for general text classification, and can be discarded. However, as demonstrated by Pang et al. (2002), Pang and Lee (2004), Hu and Liu (2004), and Riloff et al. (2003), there are some nouns and verbs that are useful sentiment indicators as well. Therefore, a clear distinction cannot be made along parts of speech categories.

To address this issue, we propose a *feature selection* scheme in which we can obtain important sentiment indicators that:

1. Do not rely on specific parts of speech classes while maintaining the focus on syntax words.

2. Separate semantic words that do not indicate sentiment while keeping nouns that do.

3. Reflect the domain for the set of documents.

By using feature selection schemes that focus on the outlined sentiment indicators as a basis for our machine learning approach, we should achieve competitive accuracy results when classifying document polarities.

The rest of this paper is organized as follows. Section 2 discusses some important work and results for SA and outlines the modelling and classification techniques used by our approach. Section 3 provides details about our feature selection methods. Our experiments and analyses are given in section 4, and conclusions and future directions are presented in section 5.

2 Related Work

2.1 Feature Selection in Sentiment Analysis

The majority of the approaches for SA involve a two-step process:

1. Identify the parts of the document that will likely contribute to *positive* or *negative* sentiments.

2. Combine these parts of the document in ways that increase the odds of the document falling into one of these two polar categories.

The simplest approach for (1) by Pang et al. (2002) is to use the most frequently-occurring words in the corpus as polarity indicators. This approach is commonly used with general text classification, and the results achieved indicate that simple document frequency cutoffs can be an effective feature selection scheme. However, this scheme picks up on many entity words that do not contain any subjectivity.

The most common approach, used by researchers such as Das and Chen (2007), starts with a manually created lexicon specific to their particular domain whereas others (Hurst and Nigam, 2004; Yi et al., 2003) attempt to craft a general-purpose opinion lexicon that can be used across domains. More recent lexicon-based approaches (Ding et al., 2008; Hu and Liu, 2004; Kim and Hovy, 2004; Riloff et al., 2003) begin with a small set of 'seed' words and bootstrap this set through synonym detection or various on-line resources to obtain a larger lexicon. However, lexicon-based approaches have several key difficulties. First, they take time to compile. Whitelaw et al. (2005) report that their feature selection process took 20 person-hours, since it involves work done by human annotators. In separate qualitative experiments done by Pang et al. (2002),

Wilson et al. (2005) and Kim and Hovy (2004), the agreement between human judges when given a list of sentiment-bearing words is as low as 58% and no higher than 76%. In addition, some words may not be frequent enough for a classification algorithm.

2.2 Topic Modelling and HMM-LDA

Topic models such as *Latent Dirichlet Allocation* (LDA) are generative models that allow documents to be explained by a set of unobserved (latent) topics. Hidden Markov Model LDA (HMM-LDA) (Griffiths et al., 2005) is a topic model that simultaneously models topics and syntactic structure in a collection of documents. The idea behind the model is that a typical word can play different roles. It can either be part of the content and serve in a semantic (topical) purpose or it can be used as part of the grammatical (syntactic) structure. It can also be used in both contexts. HMM-LDA models this behavior by inducing syntactic classes for each word based on how they appear together in a sentence using a Hidden Markov Model. Each word gets assigned to a syntactic class, but one class is reserved for the semantic words. Words in this class behave as they would in a regular LDA topic model, participating in different topics and having certain probabilities of appearing in a document. More formally, the model is defined in terms of three sets of variables and a *generative process*. Let $\mathbf{w} = \{w_1, ..., w_n\}$ be a sequence of words where each word w_i is one of V words; $\mathbf{z} = \{z_1, ..., z_n\}$, a sequence of topic assignments where each z_i is one of K topics; and $\mathbf{c} = \{c_1, ..., c_n\}$, a sequence of class assignments where each c_i is one of C classes. One class, $c_i = 1$ is designated as the 'semantic class', and the rest, the 'syntactic' classes.

Since we are dealing with a Hidden Markov Model, we require a variable representing the *transition probabilities* between the classes, given by a $C \times C$ *transition matrix* π that models transitions between classes c_{i-1} and c_i. The generative process is described as follows:

1. Sample $\theta^{(d)}$ from a Dirichlet prior $Dir(\alpha)$
2. For each word w_i in document d:
   (a) Draw $z_i \sim \theta^{(d)}$
   (b) Draw $c_i \sim \pi^{(c_i - 1)}$
   (c) If $c_i = 1$, then draw $w_i \sim \phi^{(z_i)}$, else draw $w_i \sim \phi^{(c_i)}$

where $\phi^{(z_i)} \sim Dir(\beta)$ and $\phi^{(c_i)} \sim Dir(\delta)$, both from *Dirichlet* distributions.

2.3 Text Classification Based on Maximum Entropy Modelling

Maximum Entropy Modelling (Manning and Schütze, 1999) is a framework whereby the features represent constraints on the overall model and the idea is to incorporate the knowledge that we have while preserving as much uncertainty as possible about the knowledge we do not have. The features f_i are binary functions where there is a vector x representing input elements (unigram features in our case) and c, the class label for one of the possible categories. More specifically, a feature function is defined as follows:

$$f_{i,c'}(x, c) = \begin{cases} 1 & \text{if } x \text{ contains } w_i \text{ and } c = c' \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where word w_i and category c' correspond to a specific feature.

Employing the feature functions described above, a Maximum Entropy model takes the following form:

$$P(x, c) = \frac{1}{Z} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \tag{2.2}$$

where K is the number of features, α_i is the weight for feature f_i, and Z is a normalizing constant. By taking the logarithm on both sides, we get the log-linear model:

$$\log P(x, c) = -\log Z + \sum_{i=1}^{K} f_i(x, c) \log \alpha_i \tag{2.3}$$

To classify a document, we compute $P(c|x)$ so that the c with the highest probability will be the category for the given document.

3 Feature Selection (FS) Based on HMM-LDA

3.1 Characteristics of Salient Features

To motivate our approach, we first describe criteria that are useful in selecting salient features for SA:

1. *Features should be expressive enough to add useful information to the classification process.* As discussed in section 1, the most expressive features in terms of polarity are the *modifying* words that describe an entity in a certain way. These are usually, but not restricted to, adjectives, adverbs, subjective verbs and nouns.

2. *All features together should form a broad and comprehensive viewpoint of the entire corpus.* In a corpus of many documents, some features can represent a subset of the corpus very accurately, while other features may represent another subset of the corpus. The problem arises when representing the whole corpus with a specific feature set (Sebastiani, 2002).

3. *Features should be as domain-dependent as possible.* Examples from Hurst and Nigam (2004) and Das and Chen (2007) as well as many other approaches indicate that SA is a domain-dependant task, and the final features should reflect the domain of the corpus that they are representing.

4. *Features must be frequent enough.* Rare features do not occur in many documents and make it difficult to train a machine learning algorithm. Experiments by Pang et al. (2002) indicate that having more features does not help learning, and the best accuracy was achieved by selecting features based on *document frequency*.

5. *Features should be discriminative enough.* A learning system needs to be able to pick up on their presence in certain documents for one outcome and absence in other documents for another outcome in classification.

3.2 FS Based on Syntactic Classes

Our proposed FS scheme is to utilize HMM-LDA to obtain words that, for the most part, follow the criteria we set out in subsection 3.1. We train an HMM-LDA model to give us the syntactic classes that we further combine to form our final features. Let word $w_i \in V$ where V is the vocabulary. Also let $c_j \in C$ be a class. We define $P_{c_j}(w_i)$ as the probability of word w_i in class c_j, and one class, $c_j = 1$ indicates the semantic class. Since each class (syntactic and semantic) has a probability distribution over all words, we need to select words that offer a good *representation* of the class. The representative words in each class have a much higher probability than the other words. Therefore, we can select the representative words by the *cumulative probability*. Specifically, we select the top percentage of the words in a class whereby the sum of their probabilities will be within some pre-defined range. This is necessary since there are many words in each class with low probabilities in which we are not interested (Steyvers and Griffiths, 2006). The cumulative distribution function is defined as:

$$F_j(w_i) = \sum_{P_{c_j}(w) \geq P_{c_j}(w_i)} P_{c_j}(w) \qquad (3.1)$$

Then, we can define the set of words in class c_j as:

$$W_{c_j} = \{w_i | F_j(w_i) \leq \eta\} \qquad (3.2)$$

where η is a pre-defined threshold such that $0 \leq \eta \leq 1$. Next, we define the set of words in all the syntactic classes W_{syn} as:

$$W_{syn} = \{w_i | w_i \in W_{c_j} \text{ and } c_j \neq 1\} \qquad (3.3)$$

and the set of words in the semantic class W_{sem} as:

$$W_{sem} = \{w_i | w_i \in W_{c_j} \text{ and } c_j = 1\} \qquad (3.4)$$

Since modifying words for sentiment typically fall into syntactic classes, we could use words in W_{syn} as features for SA. However, as observed by Pang et al. (2002), the best classification performance is achieved by a subset of features (typically around 2500). As a general step, we can apply a document frequency (DF) cutoff to select the most frequent features. Let $df(w_i)$ denote the document frequency of word w_i, indicating the number of documents in which w_i occurs in the corpus. Then the

resulting features selected based on df can be defined as:

$$cut(W_{syn}, \epsilon) = \{w_i | w_i \in W_{syn} \text{ and } df(w_i) \geq \epsilon\} \quad (3.5)$$

where ϵ is the minimum document frequency required for feature selection.

3.3 FS Based on Set Difference between Syntactic and Semantic Classes

The main characteristic of using HMM-LDA classes for feature selection is that the set of words in the syntactic classes and the set of words in the semantic class are not disjoint. In fact, there is quite a large overlap. In this and the next subsections, we discuss ways to remedy and even exploit this situation to get a higher level of accuracy. In the Pang et al. movie review data, there is about 35% overlap between words in the syntactic and semantic classes for $\eta = 0.9$. Our first systematic approach attempts to gain better accuracy by lowering the ratio of semantic words in the final feature set.

More formally, given the set of syntactic words W_{syn}, we can reduce the overlap with W_{sem} by doing a set difference operation:

$$W_{syn} - W_{sem} \quad (3.6)$$

This will give us all the words that are more favoured in the syntactic classes. However, as we shall see shortly, and also as we earlier speculated, by subtracting all the words in the semantic class, we are actually getting rid of some useful features. This is because (a) it is possible for the semantic class to contain words that are syntactic, and as a result are useful, and (b) there exist some semantic words that are good indicators of polarity. Therefore, we seek to 'lessen' the influence of the semantic class by cutting only a certain portion of it out, but not all of them.

For the above scheme, we outline Algorithm 1 that enables us to select features from W_{syn} by applying a percentage cutoff for W_{sem} and then doing a set difference operation. We define $top(W_{sem}, \delta)$ to be the δ% of the words with top probabilities in W_{sem}.

Note that when $\delta = 1.0$, we get the same result as $W_{syn} - W_{sem}$. In our experiments, we try a range of δ values for SA.

Algorithm 1 Syntactic-Semantic Set Difference

Require: W_{syn} and W_{sem} as input
1: $W'_{sem} = top(W_{sem}, \delta)$
2: $W_{diff} = W_{syn} - W'_{sem}$
3: $W'_{syn} = cut(W_{diff}, \epsilon)$

3.4 FS Based on Max Scores of Syntactic Features

The running theme through the HMM-LDA feature selection schemes is that if a word is highly ranked (has a high probability of occurring) in a syntactic class, we should use that word in our feature set. Moreover, if a word is highly ranked in the semantic class, we usually do not want to use that word in our feature set because the word usually indicates a frequent noun. Therefore, the desirable words are those that occur with high probability in the syntactic classes, but do not occur with high probability in the semantic class, or do not occur there at all.

To this end, we have formulated a scheme that adds such words to our feature set. For each word, we obtain its highest probability in the set of syntactic classes. Comparing this probability with the probability of the same word in the semantic class, we disregard the word if the probability in the semantic class is greater.

We define the max scores for word w_i for both the syntactic and semantic classes and describe how we select features based on the max scores in Algorithm 2.

Algorithm 2 Max Scores of Syntactic Features

Require: $c_j \in C$ where $1 \leq j \leq |C|$
1: **for all** $w_i \in V$ **do**
2: $S_{syn}(w_i) = max_{c_j \neq 1} P_{c_j}(w_i)$
3: $S_{sem}(w_i) = P_{c_1}(w_i)$
4: $W_{max} = \{w_i | S_{syn}(w_i) > S_{sem}(w_i)\}$
5: **end for**
6: $W'_{syn} = cut(W_{max}, \epsilon)$

4 Experiments

This section describes the steps taken to generate some experimental results for each scheme described in the previous section. Before we can analyze these sets of results, we take a look at some

baselines.

4.1 Evaluation

We use the corpus of 2000 movie reviews (Pang and Lee, 2004) that consists of 1000 positive and 1000 negative documents selected from on-line forums. In our experiments, we randomize the documents and split the data into 1800 for training / testing purposes and 200 as the validation set. For the 1800 documents, we run a 3-fold cross validation procedure where we train on 1200 documents and test on 600. We compare the resultant feature sets after each FS scheme using the OpenNLP[2] Maximum Entropy classifier.

Throughout these experiments, we are interested in the *classification accuracy*. This is evaluated simply by comparing the resultant class from the classifier and the actual class annotated by Pang and Lee (2004). The number of matches is divided by the number of documents in the *test* set. Thus, given an *annotated* test set $d_{test_A} = \{(d_1, o_1), (d_2, o_2), \ldots (d_S, o_S)\}$ and the classified set, $d_{test_B} = \{(d_1, q_1), (d_2, q_2), \ldots (d_S, q_S)\}$, we calculate the accuracy as follows:

$$\frac{\sum_{i=1}^{S} I(o_i = q_i)}{S} \tag{4.1}$$

where $I(\cdot)$ is the indicator function.

4.2 Baseline Results

After replicating the results from Pang et al. (2002), we varied the number of iterations per fold by using a held-out validation set 'eval'. The higher accuracy achieved suggests that the model was not fully trained after 10 iterations.

In order to compare with our HMM-LDA based schemes, we ran experiments to explore a basic POS-based feature selection scheme. In this approach, we first tagged the words in each document with POS tags and selected the most frequently-occurring unigrams that were not tagged as 'NN', 'NNP', 'NNS' or 'NNPS' (the 'noun' categories). This corresponds to **POS (-NN*)** in Table 1. Next, we tagged all the words and only selected the words that were tagged as 'JJ*', 'RB*', and 'VB*' categories (the 'syntactic' categories). The idea is to include as part of the feature set all the words that are not 'semantically oriented'. This corresponds to **POS (JJ* + RB* + VB*)** in Table 1.

Iterations	DF cutoff	POS (-NN*)	POS (JJ*+RB* +VB*)
10	0.821	0.827	0.811
25	0.836	0.831	0.824
eval	0.845	0.848	0.826

Table 1: Baseline results with a different number of iterations. Each column represents a different feature selection method.

4.3 HMM-LDA Training

Our feature selection methods involve training an HMM-LDA model on the Pang et al. corpus of movie reviews, taking the class assignments, and combining the resultant unigrams to create features for the MaxEnt classifier. Since HMM-LDA is an *unsupervised* topic model, we can train it on the entire corpus. We trained the model using the Topic Modelling Toolbox[3] MATLAB package on the 2000 movie reviews. Since the HMM-LDA model requires sentences to be outlined, we used the usual end-of-sentence markers ('.', '!', '?', ':'). The training parameters are **T = 50** topics, **S = 20** classes, **ALPHA = 1.0**, **BETA = 0.01**, and **GAMMA = 0.1**. We found that 1000 iterations is sufficient as we tracked the log-likelihood of every 10 iterations.

After training, we have both the topic assignments **z** and the class assignments **c** for each word in each of the samples.

4.4 Selecting Features Based on Syntactic Classes

In this experiment we fix $\eta = 0.9$ to get the top words in each class having a cumulative probability under 0.9. These are the *representative* words in each class which we merge into W_{syn}. Finally, we select 2500 words by the df cutoff method. This list of words is then used as features for the MaxEnt classifier. We run the classifier for 10, 25 and 'eval' number of iterations in order to compare with the baseline results.

[2] http://incubator.apache.org/opennlp/

[3] http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

Iterations	FS Based on Syntactic Features
10	0.823
25	0.839
eval	0.863

Table 2: Results for FS Based on Syntactic Classes at 10, 25 and 'eval' iterations.

At $\eta = 0.9$, there are 6,189 words in W_{syn} before we select the top 2500 using the *df* cutoff. From Table 2, we see that the accuracy has increased from 0.845 to 0.863 at the 'eval' number iterations.

In all of our experiments, we use *df* cutoff to get a manageable number of features for the classifier. This is partly based on Pang et al. (2002) and partly based on calculating the *Pearson correlation* for each class between the document frequency and word probability at $\eta = 0.9$. Since every class has a positive correlation in the range of [0.313938, 0.888160] where the average is 0.576, we can say that there is a correlation between the two values.

4.5 Selecting Features Based on Set Difference

The result for Set Difference is derived by varying the percentage of top semantic words that should be excluded in the final feature set. For example, some words in $W_{syn} \cap W_{sem}$ that have a higher probability in W_{sem} are: *'hollywod'*, *'war'*, and *'fiction'* while some words that have a higher probability in W_{syn} include: *'good'*, *'love'* and *'funny'*. The δ value is defined by the percentage of the words in W_{sem} that we exclude from W_{syn}. The results for $0.0 \leq \delta \leq 1.0$ for increments of $\delta \times |W_{sem}|$, are summarized in Table 3.

δ	FS Based on Set Difference	δ	FS Based on Set Difference
0.0	0.861	0.5	0.852
0.1	0.862	0.6	0.846
0.2	0.865	0.7	0.849
0.3	0.858	0.8	0.847
0.4	0.857	0.9	0.840
		1.0	0.831

Table 3: Results for FS Based on Syntactic-Semantic set difference method. Each row represents the accuracy achieved at a particular δ value.

From the results, we can see that as we remove more and more words from W_{sem}, the accuracy level decreases. This suggests that $W_{sem} \cap W_{syn}$ contains some important features and if we subtract W_{sem} entirely, we essentially eliminate them. At each cutoff level, we are eliminating 10% until we have eliminated the whole set. Clearly, a more fine-grained approach is needed, and that leads us to the Max-Score results.

4.6 Selecting Features Based on Max Scores

For the method based on Max Scores, we may select features that are in both W_{sem} and W_{syn} sets as long as their max scores in W_{syn} are higher than those in W_{sem}.

Iterations	FS Based on Max Scores
eval	0.875

Table 4: Result for FS Based on Max Scores.

Comparing the accuracy in Table 4 with those in the previous subsections, we can say that using the fine-grained Max-Score algorithm improves the classification accuracy. This means that iteratively removing words that have a relatively higher probability in W_{sem} compared to W_{syn} does not eliminate important words occurring in both sets, but lessens the influence of some high probability words in W_{sem}.

4.7 Discussion of the Results

For our experiments, the best accuracy is achieved by utilizing the Max-Score algorithm (outlined in subsection 3.4) after a further selection of 2500 with the *df* cutoff. As discussed in subsection 3.4, the Max-Score algorithm enables us to select words that have a higher score in W_{syn} than in W_{sem}. This approach has the dual advantage of keeping the words that are present in both W_{syn} and W_{sem} but have higher scores in W_{syn} and ignoring the words that are also present in both sets but have higher scores in W_{sem}. Ultimately, this decreases the influence of the frequent and overlapped words that have a high probability in W_{sem}.

Finally, to quantify the significance level of our best approach against the baseline methods in sub-

section 4.2, we calculated the p-values for the one-tailed t-tests comparing our best approach based on Max Scores with the DF and POS (-NN*) baselines, respectively. The resulting p-values of 0.011 and 0.014 suggest that our best approach is *significantly* better than the baseline approaches.

5 Conclusions and Future Directions

In this paper, we have described a method for feature selection based on long-range and short-range dependencies given by the HMM-LDA topic model. By modelling review documents based on the combinations of syntactic and semantic classes, we have devised a method of separating the topical content that describes the *entities* under review from the opinion context (given by sentiment *modifiers*) about that entity in each case. By grouping all the sentiment modifiers for each entity in a document, we are selecting the features that are intuitively in line with the outlined characteristics of salient features for SA (see subsection 3.1). This is backed up by our experiments where we achieve competitive results for document polarity classification.

One avenue for future development of this framework could include identifying and extracting *aspects* from a review document. So far, we have not identified aspects from the entities, choosing instead to classify a document as a whole. However, this framework can be readily applied to extract relevant (most probable) aspects using the LDA topic model and then restrict the syntactic modifiers to the range of sentences where an aspect occurs. This would give us an *unsupervised* aspect extraction scheme that we can combine with a classifier to predict polarities for each aspect.

References

Sanjiv R. Das and Mike Y. Chen. 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the Web. *Management Science*, 53(9):1375–1388.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM)*.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of AAAI*, pages 755–760.

Matthew Hurst and Kamal Nigam. 2004. Retrieving topical sentiments from online document collections. In *Document Recognition and Retrieval XI*, pages 27–34.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 271–278.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 25–32.

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

Mark Steyvers and Tom Griffiths. 2006. Probabilistic topic models. In T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning.* Laurence Erlbaum.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, pages 625–631. ACM.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 347–354.

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.