Pruning Non-Informative Text Through Non-Expert Annotations to Improve Aspect-Level Sentiment Classification

Ji Fang
Palo Alto Research Center
Ji.Fang@parc.com

Bob Price
Palo Alto Research Center
Bob.Price@parc.com

Lotti Price
Palo Alto Research Center
Lotti.Price@parc.com

Abstract

Sentiment analysis attempts to extract the author's sentiments or opinions from unstructured text. Unlike approaches based on rules, a machine learning approach holds the promise of learning robust, high-coverage sentiment classifiers from labeled examples. However, people tend to use different ways to express the same sentiment due to the richness of natural language. Therefore, each sentiment expression normally does not have many examples in the training corpus. Furthermore, sentences extracted from unstructured text (e.g., I filmed my daughter's ballet recital and could not believe how the auto focus kept blurring then focusing) often contain both informative (e.g., the auto focus kept blurring then focusing) and extraneous non-informative text regarding the author's sentiment towards a certain topic. When there are few examples of any given sentiment expression, extraneous non-sentiment information cannot be identified as noise by the learning algorithm and can easily become correlated with the sentiment label, thereby confusing sentiment classifiers. In this paper, we present a highly effective procedure for using crowd-sourcing techniques to label informative and non-informative information regarding the sentiment expressed in a sentence. We also show that pruning non-informative information using non-expert annotations during the training phase can result in classifiers with better performance even when the test data includes non-informative information.

1 Introduction

Noise in training data can be derived either from noisy labeling or from noisy features. It has been shown that labeling quality is one of the important factors that impacts the performance of a learned model, and that this quality can be improved by approaches such as using multiple labelers (Sheng et al., 2008). However, noisy features can be an inherent characteristic for some text mining tasks, and it is unclear how they should be handled.

For example, sentiment analysis/opinion mining from unstructured user generated content such as online reviews and blogs often relies on learning sentiments from word-based features extracted from the training sentences and documents (Pang et al., 2002; Dave et al., 2003; Kim and Hovy, 2005). However, not all words in the training data carry information about sentiment. For example, in sentence (1),

(1)*I filmed my daughter's ballet recital and could not believe how the auto focus kept blurring then focusing.*

although words such as *auto focus, blurring* and *focusing* are informative for learning sentiment regarding the auto focus capability of the camera, words such as *film, daughter* and *ballet recital* are not informative for that type of sentiment, and they form noise if included as training data.

If the training data contain a lot of examples such as (2) in which words such as *film, daughter* and *ballet recital* also appear, but the sentence is not labelled as invoking sentiment regarding auto focus, a machine learning algorithm might learn

Beijing, August 2010
Proceedings of the 2nd Workshop on "Collaboratively Constructed Semantic Resources", Coling 2010, pages 37–45,

that such words are not informative for sentiment classification.

(2)*I filmed my daughter's ballet recital and could not believe how good the picture quality was.*

However, due to the richness of natural language, people tend to use different ways to describe a similar event or to express a similar opinion. Consequently, repeated use of the same expression is not common in the training data for sentiment classification. Note that this difficulty cannot be simply overcome by increasing the size of the training data. For example, a search on the completely natural phrase "I filmed my daughter's ballet recital" in Google and Bing returns the same exact sentence as shown in (1). In other words, there appears to be only one sentence containing that exact phrase, which implies that even if we use the entire web as our training data set we would not find an example such as (2) to help the learning algorithm to determine which feature words in (1) are informative and which are not. Therefore, data sparsity is an inherent problem for a task such as sentiment analysis, and if we adopt the bag-of-words approach for sentiment classification (Pang et al., 2002), which uses the words that appear in sentences as training features, our training data will unavoidably include many noisy non-informative features.

This paper presents a crowd-sourcing technique to identify and prune the non-informative features. We explore the effect of using non-expert annotations to gain low-noise training data for sentiment classification. We show that the cleaner training data obtained from non-expert annotations significantly improve the performance of the sentiment classifier. We also present evidence that this improvement is due to reduction in confusion between classes due to noise words.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 describes our approach for pruning non-informative features. Section 4 presents an empirical study on the effect of training on informative features in the domain of sentiment analysis. Conclusions are summarized in Section 5.

2 Related Work

Feature selection in the domain of sentiment analysis has focused on the following issues.

a) Should word-based features be selected based on frequency or presence?

It has been shown that compared to word frequency, word presence is a better sentiment indicator (Pang et al., 2002; Wiebe et al., 2004; Yang et al., 2006). In other words, unlike in other domains such as topic classification where the frequency of words provides useful information regarding the topic class, sentiment information is not normally indicated by the frequency of certain words, because people are unlikely to repeatedly use the same word or phrase to express an opinion in one document. Instead, Researchers (Pang et al., 2002) found that selecting features based on word presence rather than word frequency leads to better performance in the domain of sentiment analysis.

b) Which are more useful features: unigrams, higher-order n-grams or syntactically related terms?

This issue seems to be debatable. While some researchers (Pang et al., 2002) reported that unigrams outperform both bigrams as well as the combination of unigrams and bigrams in classifying movie reviews based on sentiment polarity, some others (Dave et al., 2003) reported the opposite in some settings.

Similarly, some (Dave et al., 2003) found syntactically related terms are not helpful for sentiment classification, whereas others (Gamon, 2004; Matsumoto et al., 2005; Ng et al., 2006) found the opposite to be true.

c) In terms of part-of-speech, which types of words are more useful features?

Adjectives and adverbs are commonly used as features for sentiment learning (Mullen and Collier, 2004; Turney, 2002; Whitelaw et al., 2005). However, more recent studies show that all content words including nouns, verbs, adjectives and adverbs are useful features for sentiment analysis (Dillard, 2007).

Regardless of which types of features are used, these traditional approaches are still inherently noisy in the sense that non-informative

words/features within each sentence are included as described in Section 1. As far as we are aware, this is an issue that has not been addressed.

The closest works are Riloff et al. (Riloff and Wiebe, 2003) and Pang et al. (Pang et al., 2002)'s work. Riloff et al. explored removing the features that are subsumed in other features when a combination of different types of features such as unigrams, bigrams and syntactically related terms is used. Pang et al. speculated that words that appear at certain positions in a movie review are more informative for the overall opinion reflected in that review. However, according to Pang et al., for the task of predicting the overall polarity of a movie review, training on word features assumed to be more informative resulted in worse performance than training on all word features appearing in the reviews.

Our approach is different in that we try to identify and prune non-informative word features at the sentence level. We focus on identifying which portion of the sentence is informative for sentiment classification. We then completely remove the non-informative portion of the sentence and prevent any terms occurring in that portion from being selected as feature vectors representing that sentence. Note that the classification of words as non-informative is not related to their positions in a sentence nor to their frequency count in the training corpus. Instead, whether a word is informative depends purely on the semantics and the context of the sentence. For example, the word *big* would be non-informative in (3), but informative in (4).

(3)*That was a big trip, and I took a lot of pictures using this camera.*

(4)*This camera has a big LCD screen.*

Unlike the traditional approach of using expert annotation to identify the non-informative text in a sentence, we instead use non-expert annotations without external gold standard comparisons. There have been an increasing number of experiments using non-expert annotations for various Natural Language Processing (NLP) tasks. For example, Su et al. (Su et al., 2007) use non-expert annotations for hotel name entity resolution. In (Nakov, 2008), non-expert annotators generated paraphrases for 250 noun-noun compounds, which were then used as the gold standard data for evaluating an automatic paraphrasing system. Kaisser and Lowe (Kaisser and Lowe, 2008) also use non-experts to annotate answers contained in sentences and use the annotation results to help build a question answering corpus. Snow et al. (Snow et al., 2008) reported experiments using non-expert annotation for the following five NLP tasks: affect recognition, word similarity, recognizing textual entailment, event temporal ordering, and word sense disambiguation.

This paper presents a study of using non-expert annotations to prune non-informative word features and training a sentiment classifier based on such non-expert annotations. The following section describes our approach in detail.

3 Non-Informative Feature Pruning Through Non-Expert Annotations

To prune the non-informative features, a traditional approach would be to hire and train annotators to label which portion of each training sentence is informative or non-informative. However, this approach is both expensive and time consuming. We overcome these issues by using crowdsourcing techniques to obtain annotations from untrained non-expert workers such as the ones on the Amazon Mechanical Turk (AMT) platform[1]. To illustrate our approach, we use an example for sentiment analysis below.

The key to our approach relies on careful design of simple tasks or HITs that can elicit the necessary information for both labeling the sentiment information and pruning the non-informative text of a sentence. These tasks can be performed quickly and inexpensively by untrained non-expert workers on the AMT platform. We achieved this goal by designing the following two experiments.

Experiment 1 asks the workers to judge whether a sentence indicates an opinion towards a certain aspect of the camera, and if so, whether the opinion is positive, negative or neutral. For example, the proper annotations for sentence (5) would be as shown in Figure 1.

[1]This is an online market place that offers a small amount of money to people who perform some "Human Intelligence Tasks" (HITs). https://www.mturk.com/mturk/welcome

(5) *On my trip to California, the camera fell and broke into two pieces.*

Figure 1: Experiment 1

Feature Name	Not Invoked	Positive	Negative	Neutral
Construction Quality	○	○	◎	○
Picture Quality	◎	○	○	○
Battery Life	◎	○	○	○
...				

We randomly selected 6100 sentences in total for this experiment from the Multi-Domain Sentiment Dataset created by Blitzer et al. (Blitzer et al., 2007). Each sentence was independently annotated by two AMT workers. Each annotation consisted of a sentence labeled with a camera aspect and a sentiment toward that aspect.

One unique characteristic of Experiment1 is that it makes the detection of unreliable responses very easy. Because one sentence is unlikely to invoke many different aspects of cameras, an annotation is thus suspicious if many aspects of camera are annotated as being invoked. Figure 2 and Figure 3 illustrate the contrast between a normal reliable response and a suspicious unreliable response.

Due to this favorable characteristic of Experiment 1, we did not have to design a qualification test. We approved all of the assignments; however we later filtered out the detected suspicious responses, which accounted for 8% of the work. Even though we restricted our AMT workers to those who have an approval rate of 95% or above, we still found 20% of them unreliable in the sense that they provided suspicious responses.

Given our ability to detecting suspicious responses, we believe it is very unlikely for two reliable AMT workers to annotate any given sentence exactly the same way merely by chance. Therefore, we consider an annotation to be gold when both annotators marked the same sentiment toward the same aspect. We obtained 2718 gold-standard annotations from the reliable responses. We define the agreement rate of annotations as follows.

$$AgreementRate = \frac{NumberofGoldAnnotations \times 2}{TotalNumberofAnnotations}. \quad (1)$$

Based on this measure, the agreement rate of the AMT workers in this study is 48.4%.

We held randomly selected 587 gold annotated sentences as our test set, and used the remaining 2131 sentences as our training sentences. To prune the non-informative text from the training sentences, we put the 2131 sentences through Experiment 2 as described below.

Experiment 2 asks the workers to point out the exact portion of the sentence that indicates an opinion. The opinion and its associated feature name are displayed along with the sentence in which they appear. Such information is automatically generated from the results derived from Experiment 1. An example of Experiment 2 is given in Figure 4.

Figure 4: Experiment 2

Please examine the sentence:

One thing I have to mention is that the battery door keeps falling off.

Opinion: A **Negative** opinion toward the feature **Construction Quality**.

Remove parts of the sentence that are not relevant to the opinion and put the result below. Do not rewrite the sentence; only remove words from it.

The expected answer for this example is *the battery door keeps falling off.*

Using this method, we can remove the non-informative part of the sentences: *One thing I have to mention is that* and prevent any of the words in that part from being selected as our training features.

Experiment 2 requires the workers to enter or copy and paste text in the box, and 100% of the workers did it. In our sentiment classification experiment described below, we used all of the results without further filtering.

We paid $0.01 for each assignment in both experiments, and we acquired all of the annotations in one week's time with a total cost of $215, including fees paid to Amazon. Our pay rate is about $0.36/hour. For Experiment 1 alone, if we adopted a traditional approach and hired two annotators, they could likely complete the annotations in five 8-hour days. Using this approach, the cost for Experiment 1 alone would be $1200, with a rate of $15/hour. Therefore, our approach is both cheaper and faster than the traditional approach.

Figure 2: Reliable Response

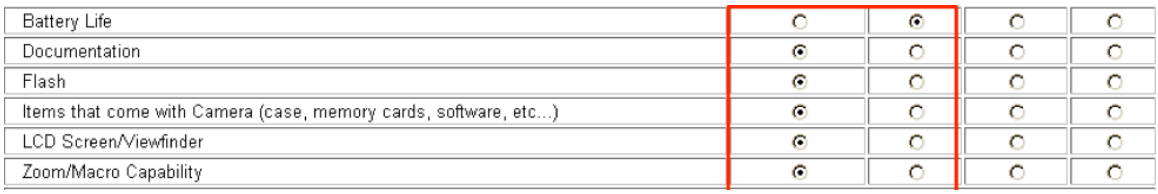
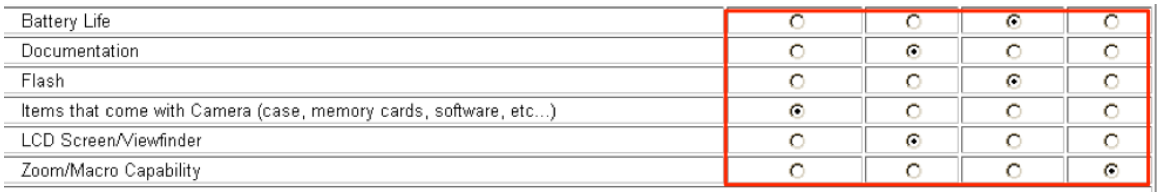


Figure 3: Unreliable Response



Having described our crowd-souring based approach for pruning the non-informative features, we next present an empirical study on the effect of training on informative features.

4 Pruning Non-Informative Features for Sentiment Classification

We conducted an experiment on sentiment classification in the domain of camera reviews to test the effect of pruning non-informative features based on AMT workers' annotations.

In our experiment, we select the Nouns, Verbs, Adjectives and Adverbs as our unigram features for training. We define non-informative features as the four types of words occurring in the non-informative portion of the training sentence; namely, the portion that does not mention any aspect of the camera or associated sentiment. For example, for a training sentence such as (1) (repeated below as (6)), training on all features would select the following words: [*film, daughter, ballet, recital, not-believe*[2]*, auto, focus, kept, blurring, focusing*].

(6) *I filmed my daughter's ballet recital and could not believe how the auto focus kept blurring then focusing.*

By contrast, pruning non-informative features would yield a shorter list of selected words: [*auto, focus, kept, blurring, focusing*].

In our experiment, we compare the performance of the classifier learned from all of the Nouns, Verbs, Adjectives and Adverbs in the sentences with the one learned from these word types occurring only in the informative part of the sentence. When the training set contains all of the feature words, we refer to it as the All-Features-Set. When the non-informative features are pruned, the training set contains only the informative feature words, which we refer to as the Informative-Features-Set.

All of the feature words are stemmed using the Porter Stemmer (Porter, 1980). Negators are attached to the next selected feature word. We also use a small set of stop words[3] to exclude copulas and words such as *take*. The reason that we choose these words as stop words is because they are both frequent and ambiguous and thus tend to have a negative impact on the classifier.

All of our training and test sentences are annotated through crowd-sourcing techniques as described in the last section. In our experiment we use 2131 sentences in total for training and 587 sentences for hold-out testing. The non-informative part of the test sentences are not removed. The experiment results and implications are discussed in detail in the following subsections.

[2]See below for the description regarding how we handle negation.

[3]The stop words we use include copulas and the following words: *take, takes, make, makes, just, still, even, too, much, enough, back, again, far, same*

4.1 Aspect:Polarity Classification Using SVM

In this experiment, the task is to perform a 45 way sentiment classification. These 45 classes are derived from 22 aspects related to camera purchases such as *picture quality, LCD screen, battery life and customer support* and their associated polarity values *positive* and *negative*, as well as a class of *no opinion* about any of the 22 aspects. An example of such a class is *picture quality: positive*. The classifier maps each input sentence into one of the 45 classes.

One of the approaches we tested is to train the classifier based on the All-Features-Set derived from the original raw sentences. We refer to this as "All Features". The other approach is to learn from the Informative-Features-Set derived from the sentences with the non-informative portion removed by the AMT workers. We refer to this as "Informative Features". The experiment is conducted using SVM algorithm implemented by Chang et al. (Chang and Lin, 2001). We use linear kernel type and use the default setting for all other parameters.

The classification accuracy is defined as follows.

$$Accuracy = \frac{NumberofSentencesCorrectlyClassified}{TotalNumberofSentences}. \quad (2)$$

The experiment results in terms of classification accuracy are shown in Table 1.

Table 1: Classification Accuracy

All Features	Informative Features
41.7%	**45.8%**

In this experiment, pruning the non-informative features improves the accuracy by more than 4%. This improvement is statistically significant by a one-tailed sign test at $p = 0.15$. Training on the informative features also consistently improves the classification accuracy when we vary the size of the training data as illustrated by the Figure 5[4].

[4]To demonstrate the learning curve, we experimented with the use of different percentages of the training sentences while always testing on the same 587 test sentences. When the percentage of the training sentences used is less than 100%, we randomly pick that percentage of training sentences until the test accuracy converges.

Figure 5: Learning Curve

A salient characteristic of this experiment is that the training data tend to be very sparse for two reasons. First, the number of classes is large, which means that the number of training examples for each class will be fewer. As shown in Table 2, 24 out of the 45 classes have fewer than 30 training examples, which is an indication of how sparse the training data is. Second, as shown in Section 1, people tend to use different ways to express the type of sentiments that we aim to learn in this experiment. Therefore, it is difficult to collect repeated training examples and this difficulty cannot be simply overcome by increasing the size of the training data. This data sparsity means that it is difficult for the SVM to learn which feature words are non-informative noise.

Table 2: Class Distribution in Experiment 1

Number of Classes	Number of Training Sentences
6	fewer than 10
14	fewer than 20
24	fewer than 30
33	fewer than 50
41	fewer than 100
4	more than 100

4.2 Automatic Feature Selection vs. Pruning by AMT Workers

As shown in the previous subsection, pruning non-informative word features using non-expert annotations can significantly improve the performance of the sentiment classifier. Can we achieve the same improvement by using automatic feature selection algorithms?

We tried three widely used feature selection techniques LR(Likelihood Ratio), WLLR(Weighted Log-Likelihood Ratio) (Nigam et al., 2000; Ng et al., 2006) and MI(Mutual Information) and applied them to the original raw training data. We found that in general, the fewer

the feature words selected by these algorithms, the worse the classifier performs. The classifier performed the best when using all of the available feature words. In other words, automatic feature selection offered no benefit. Table 3 shows the results of using these three automatic feature selection techniques as well as the results of not performing automatic feature selection. The threshold for the LR algorithm was set to be 5; the threshold for the WLLR algorithm was set to be 0.005; and the threshold for the MI algorithm was set to be 2000 (using the top 2000 ranked features out of a total of 3279 features).

Table 3: Automatic Feature Selection Results

No Feature Selection	LR	WLLR	MI
41.7%	35.4%	40.2%	41.1%

This result is not surprising given the data sparsity issue in our experiment. Traditional feature selection methods either try to remove correlated features which can cause havoc for some methods or to prune out features uncorrelated with labels to make learning more efficient. However, we have sparse data so correlations calcuated are very unstable - if a feature appears once with a label what can we conclude? So the same properties that cause difficulties for the learner cause problems for feature selection techniques as well.

To summarize, pruning non-informative word features using non-expert annotations can significantly improve the performance of the sentiment classifier even when the test data still contain non-informative features. We believe this is because pruning non-informative feature words based on human knowledge leads to better training data that cannot be achieved by using automatic feature selection techniques. The subsection below compares the two sets of training sentences we used in this experiment: one comprises the original raw sentences and the other comprises sentences with the non-informative text removed. We show that our approach of pruning non-informative text indeed leads to a better set of training data.

4.3 Comparison of Training Data Before and After the Feature Pruning

Our assumption is that training data is better if data belonging to closer classes are more similar and data belonging to further classes are more different. In our sentiment classification experiment, an example of two very close classes are *battery life: positive* and *battery life: negative*. An example of two very different classes are *battery life: positive* and *auto focus: negative*. The more similar the training data belonging to closer classes and the more dissimilar the training data belonging to different classes, the more accurate the classifier can predict the involved camera aspect, which in turn should lead to improvements on the overall classification accuracy.

To test whether the pruned text produced better training data than the original text, an adjusted cosine similarity measure was used. Note that our measurement can only reflect partial effects of AMT workers' pruning, because our measure is essentially term frequency based, which can reflect similarity in terms of topic (camera aspects in our case) but not similarity in terms of polarity (Pang et al., 2002). Nevertheless, this measurement highlights some of the impact resulting from the pruning.

To compare training data belonging to any two classes, we produce a tf-idf score for each word in those two classes and represent each class as a vector containing the tf-idf score for each word in that class. Comparing the similarity of two classes involves calculating the adjusted cosine similarity in the following formula.

$$similarity = \frac{A \cdot B}{||A||||B||}. \quad (3)$$

A and B in the above formula are vectors of tf-idf scores, whereas in the standard cosine similarity measure A and B would be vectors containing tf scores. The motivation for using tf-idf scores instead of the tf scores is to reduce the importance of highly common words such as *the* and *a* in the comparison. The similarity score produced by this formula is a number between 0 and 1; 0 being no overlap and 1 indicating that the classes are identical. Word stemming was not used in this experiment.

We compared similarity changes in two situations. First, when two classes share the same aspect; this involves comparison between 22 class pairs such as *battery life: positive* vs. *battery life: negative*. Second, when two classes share different aspects; for example, *battery life: positive* vs. *auto focus: negative* and *battery life: positive* vs. *auto focus: positive*. In this situation, we compared the similarity changes in 903 class pairs. If pruning the non-informative text does indeed provide better training data, we expect similarity to increase in the first situation and to decrease in the second situation after the pruning. This is precisely what we found; our finding is summarized in Table 4.

Table 4: Average Similarity Changes in the Pruned Training Data

Same aspect	Different aspect
+0.01	-0.02

In conclusion, AMT workers, by highlighting the most pertinent information for classification and allowing us to discard the rest, provided more useful data than the raw text.

5 Conclusions

To summarize, we found that removing the non-informative text from the training sentences produces better training data and significantly improves the performance of the sentiment classifier even when the test data still contain non-informative feature words. We also show that annotations for both sentiment classes and sentiment-informative texts can be acquired efficiently through crowd-sourcing techniques as described in this paper.

6 Acknowledgments

We thank Prateek Sarkar, Jessica Staddon and Bi Chen for insightful discussions and comments. We thank the anonymous reviewers' helpful suggestions and feedback. We would also like to thank Jason Kessler for implementing part of the sentiment analysis algorithm and the Amazon Mechanical Turk experiments.

References

Sheng, Victor S., Provost, Foster, and Ipeirotis, Panagiotis G.. 2008. *Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers*. KDD 2008 Proceedings 614-622.

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

Dave, Kushal, Lawrence, Steve, and Pennock, David M.. 2003. *Mining the peanut gallery: Opinion extraction and semantic classification of product reviews*. Proceedings of WWW 519-528.

Kim, Soo–Min and Hovy, Eduard. 2005. *Identifying opinion holders for question answering in opinion texts*. Proceedings of the AAAI Workshop on Question Answering in Restricted Domains.

Wiebe, Janyce M. , Wilson, Theresa , Bruce, Rebecca , Bell, Matthew and Martin, Melanie. 2004. *Learning subjective language*. *Computational Linguistics*, 30(3):277-308.

Yang, Kiduk , Yu, Ning , Valerio, Alejandro and Zhang, Hui. 2006. *WIDIT in TREC-2006 Blog track*. Proceedings of TREC.

Gamon, Michael. 2004. *Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis*. Proceedings of the International Conference on Computational Linguistics (COLING).

Matsumoto, Shotaro, Takamura, Hiroya and Okumura, Manabu. 2005. *Sentiment classification using word sub-sequences and dependency sub-trees*. Proceedings of PAKDD05, the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining.

Ng, Vincent, Dasgupta, Sajib and Arifin, S. M. Niaz. 2006. *Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews*. Proceedings of the COLING/ACL Main Conference Poster Sessions 611-618.

Mullen, Tony and Collier, Nigel. 2004. *Sentiment analysis using support vector machines with diverse in-formation sources*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 412-418.

Turney, Peter. 2002. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. Proceedings of the Association for Computational Linguistics (ACL) 417-424.

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

Dillard, Logan. 2007. *I Can't Recommend This Paper Highly Enough: Valence-Shifted Sentences in Sentiment Classification*. Master Thesis.

Riloff, Ellen and Wiebe, Janyce. 2003. *Learning extraction patterns for subjective expressions*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Su, Qi, Pavlov, Dmitry, Chow, Jyh-Herng and Baker, Wendell C.. 2007. *Internet-Scale Collection of Human- Reviewed Data*. Proceedings of WWW-2007.

Nakov, Preslav. 2008. *Paraphrasing Verbs for Noun Compound Interpretation*. Proceedings of the Workshop on Multiword Expressions, LREC-2008.

Kaisser, Michael and Lowe, John B.. 2008. *A Re-search Collection of QuestionAnswer Sentence Pairs*. Proceedings of LREC-2008.

Snow, Rion, O'Connor, Brendan, Jurafsky, Daniel and Ng, Andrew Y. 2008. *Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks*. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Blitzer, John, Dredze, Mark, Biographies, Fernando Pereira., Bollywood, Boom-boxes and Blenders. 2007. *Domain Adaptation for Sentiment Classification*. Proceedings of the Association for Computational Linguistics (ACL).

Porter, M.F.. 1980. *An algorithm for suffix stripping*. Program.

Chang, Chih-Chung and Lin, Chih-Jen. 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Nigam, K., McCallum, A.K., Thrun, S., and Mitchell, T.. 2000. *Text Classification from labeled and unlabeled documents using em*. Machine Learning 39(2-3) 103-134.