Compact Features for Sentiment Analysis

Lisa Gaudette[1] and Nathalie Japkowicz[1]

School of Information Technology & Engineering
University of Ottawa
Ottawa, Ontario, Canada
{lgaud082,njapkow}@uottawa.ca

Abstract. This work examines a novel method of developing features to use for machine learning of sentiment analysis and related tasks. This task is frequently approached using a "Bag of Words" representation – one feature for each word encountered in the training data – which can easily involve thousands or tens of thousands of features. This paper describes a set of compact features developed by learning scores for words, dividing the range of possible scores into a number of bins, and then generating features based on the distribution of scored words in the document over the bins. This allows for effective learning of sentiment and related tasks with 25 features; in fact, performance was very often slightly better with these features than with a simple bag of words baseline. This vast reduction in the number of features reduces training time considerably on large datasets, and allows for using much larger datasets than previously attempted with bag of words approaches, improving performance.

1 Introduction

Sentiment analysis is the problem of learning opinions from text. On the surface, sentiment analysis appears similar to text categorization by topic, but it is a harder problem for many reasons, as discussed in Pang & Lee's 2008 survey [14]. First and foremost, with text categorization, it is usually much easier to extract relevant key words, while sentiment can be expressed in many ways without using any words that individually convey sentiment. In topic classification there are undoubtedly red herrings, such as the use of analogies and metaphor, but if a word associated with a given domain is mentioned frequently, it is usually related (although not necessarily the most relevant). However, in sentiment analysis there are many examples of "thwarted expectations"[1] and comparison to an entity with opposing sentiment[2] such that a positive review can easily have many negative words and vice versa [14]. In addition, words that convey positive sentiment in one domain may be irrelevant or negative in another domain, such as the word *unpredictable*, which is generally positive when referring to the plot of a book or a movie but negative when referring to an electronic device.

[1] "I was expecting this movie to be great, but it was terrible"

[2] "I loved the first movie, but this sequel is terrible"

The Bag of Words (BOW) representation is commonly used for machine learning approaches to text classification problems. This representation involves creating a feature vector consisting of every word seen (perhaps some minimum number of times) in the training data and learning based on the words that are present in each document, sentence, or other unit of interest. However, this approach leads to a very large, sparse feature space, as words are distributed such that there are a small set of very frequent, not very informative words, and a great deal of individually rarer words that carry most of the information in a sentence, roughly following Zipf's Law [9]. Thousands of features are required to adequately represent the documents for most text classification tasks.

Another approach is to learn scores for words, and then use these words to classify documents based on the sum or average of the scores in a document and some threshold. While this type of approach is useful, bag of words based approaches generally perform better, although the two approaches can be combined together through meta classifiers or other approaches to improve on either approach individually.

This research proposes a novel method of combining machine learning with word scoring through condensing the sparse features of BOW into a very compact "numeric" representation by using the distribution of the word scores in the document. This approach allows for much smaller feature sets, and much faster processing of large data sets.

2 Related Work

This paper combines ideas from two different basic approaches to sentiment analysis. The first is to use a bag of words feature set and train a machine learning classifier such as a Support Vector Machine (SVM) using the BOW features, such as in [13]. The second approach is to learn scores for words and score documents based on those scores, such as in [3]. Some previous attempts to combine these two approaches include combining results from both systems with a meta classifier such as in [11], [7], and [1], and weighting bag of words features by a score, such as the use of TF/IDF in [10]. While there are many techniques to improve on the basic idea of BOW through refining the features or combining it with other approaches, it remains a good basic approach to the problem.

3 Approach to Generating Compact Features

The approach used here involves 3 steps. The first step is to learn scores for the words, while the second is to represent the documents in terms of the distribution of those word scores. Finally, we run a machine learning algorithm on the features representing the distribution of the word scores. We refer to our features as "Numeric" features.

3.1 Learning Word Scores

The first step to this approach involves learning word scores from the text. We initially considered three different supervised methods of scoring words, and found that Precision performed best.

This method was inspired by its use in [16] for extracting potential subjective words. It represents the proportion of occurrences of the word which were in positive documents, but does not account for differences in the number of words in the sets of positive and negative documents. This produces a value between 0 and 1.

$$precision = \frac{wP}{wP + wN} \tag{1}$$

wP, wN the number of occurrences of word w in positive (negative) documents

In order to calculate the precision, we first go through the training data and count the number of positive and negative instances of each word. We then compute the scores for each word. As a word which appears very few times could easily appear strongly in one category by chance, we chose to only use words appearing at least 5 times. This produces a list of word scores customized to the domain of interest.

3.2 Example of Scoring Words

Consider scoring the word "good" from a very small corpus consisting of the following four sentences:

Positive	Negative
The acting was *good.*	I thought this movie would be *good*, but it was awful.
This movie is very *good.*	This movie is terrible.

"Good" appears 2 times in positive documents, and 1 time in negative documents, for a precision of $\frac{2}{2+1} = 0.667$.

3.3 Generating Features from Scored Words

In order to generate features from these scored words, we first divide the range of possible scores into a number of "bins", representing a range of word scores. We then go through each document, look up the score for each word, and increment the count of its corresponding bin. After we have counted the number of words in each bin, we normalize the count by the number of scored words in the document, such that each bin represents the percentage of the words in the document in its range of scores.

Example of Generating Features. This section shows an example of scoring a document after we have scored the words, assuming 10 bins and precision word scores ranging from 0 to 1. Figure 1a shows the preprocessed text of the review with word scores, while Figure 1b shows the results of counting the number of words in each bin, and then normalizing those counts based on the number of scored words in the document to generate the features we use for machine learning.

0.460	0.503	0.545	0.555	0.576
i	have	always	been	very
0.850	0.526	0.497	0.449	0.351
pleased	with	the	sandisk	products
0.460	0.403	0.898	0.568	0.465
i	would	highly	recommend	them

(a) Review Preprocessed and Annotated with Word Scores

Range	Count	Feature
0.00-0.10	0	0.000
0.10-0.20	0	0.000
0.20-0.30	0	0.000
0.30-0.40	1	0.067
0.40-0.50	6	0.400
0.50-0.60	6	0.400
0.60-0.70	0	0.000
0.70-0.80	0	0.000
0.80-0.90	2	0.133
0.90-1.00	0	0.000

(b) Numeric Features Generated from Review

Fig. 1: Generating Features from a 5 star review

After going through this process for a set of documents, we have a set of numeric features based on the distribution of the word scores that is much more compact than the bag of words representation and can be used as input to a machine learning algorithm.

4 Selecting Parameters

There are three main options to this approach – the method for scoring the words, the number of bins to use, and the machine learning algorithm to use. We used two basic datasets to select these options – the reviews of Steve Rhodes from [13] and the 2000 review, balanced, Electronics dataset from [2]. We used these datasets in terms of both ordinal and binary problems, for a total of 4 distinct problems. We examined the effect of three different scoring methods, varying the number of bins, and varying the classifier using a variety of classifiers as implemented in the WEKA machine learning system [17]. For more details on these experiments, refer to [4].

While we do not have space to present all of the details here, we found that the precision scoring method performed best on most datasets by a small margin, but that all scoring methods were close. The number of bins only affected performance by a very small amount given enough bins – some datasets performed very well with as few as 10 bins, while others needed 25, and using more bins had no consistent effect on performance beyond that point. The SMO implementation of SVM from WEKA performed well, while we also found that BayesNet performed nearly as well and was much faster, particularly in the ordinal case.

The experiments using the word score based features all use precision scoring with 25 bins. In the Binary case, they use the SMO implementation of SVM from WEKA, with default settings except for the option of fitting logistic models to the outputs for Binary problems. For ordinal problems, the BayesNet classifier is used instead. The BOW baseline classifiers are all constructed using SMO with default settings, as SVM has been shown to perform well in previous work. BayesNet did not perform well using the BOW representation.

5 Notes on Evaluation

Many authors working in this domain have simply reported accuracy as an evaluation metric, which has problems in even the binary case as noted by [15] and others since. For the ordinal problem, we will use Mean Squared Error (MSE), as it was shown in [5] to be a good measure, while for the binary problem we include AUC. We include accuracy in places to compare with previous work. Where multiple runs were feasible we use 10x10 fold cross validation.

6 Experiments

In order to evaluate the feasibility of this method, we test it across a range of datasets. In all cases, we compare the results to an SVM classifier using BOW features, which represents a solid baseline approach; where available, we also provide results obtained by the authors who introduced the datasets. We evaluate both classifiers using only unigrams that appear at least 5 times in the training data.

Times reported include all time taken to read in and process the documents into the respective representations, as well as the time to train and test the classifier, averaged over all folds where multiple runs were performed.

We have selected a range of datasets on which to evaluate this approach. We have both “document” level datasets, representing units that are (at least usually) several sentences or more long, and “sentence” level datasets, representing units of about one sentence (although sometimes a phrase, or two or three sentences). We also have a contrast between sometimes poorly written online user reviews of products and more professionally written movie reviews. We have one set of datasets which contain an order of magnitude more documents than the others on which to examine the effects of adding more documents. Finally, we have one dataset that is for a slightly different problem than the others – subjectivity detection rather than sentiment analysis.

Table 1: Amazon reviews, BOW vs. Numeric, Accuracy, AUC, and Time

Dataset	Type	Accuracy	AUC	Time(mm:ss)
Electronics	Numeric	0.801 ± 0.005	0.874 ± 0.004	0:01.0
(0.844)[a]	BOW	0.791 ± 0.005	0.791 ± 0.005	0:22.2
DVD	Numeric	0.797 ± 0.005	0.865 ± 0.005	0:01.4
(0.824)	BOW	0.775 ± 0.006	0.776 ± 0.006	0:37.8
Book	Numeric	0.768 ± 0.005	0.839 ± 0.005	0:01.4
(0.804)	BOW	0.754 ± 0.006	0.754 ± 0.006	0:42.9
Kitchen	Numeric	0.814 ± 0.006	0.896 ± 0.004	0:01.0
(0.877)	BOW	0.809 ± 0.005	0.809 ± 0.005	0:23.5
All Data[b]	Numeric	0.796 ± 0.003	0.874 ± 0.002	0:04.6
	BOW	0.791 ± 0.002	0.791 ± 0.002	10:00.1
Average[c]	Numeric	0.795 ± 0.005	0.869 ± 0.005	0:04.9
	BOW	0.782 ± 0.006	0.782 ± 0.006	2:06.5
Majority[d]		0.474	0.500	

[a] Accuracy in [2]

[b] One classifier trained on all datasets. BOW classifier is 2x10 CV, all other classifiers 10x10 CV

[c] Average performance of the individual classifiers over all datasets and total time to train the individual classifiers

[d] The results for the majority classifier are the same for all datasets given the same seed for the split of the data into folds

6.1 Small Amazon Reviews

This dataset consists of online user reviews across 4 categories and was used in [2]. As shown in Table 1, the numeric features are always slightly more accurate with substantially higher AUC, while also being considerably faster than the BOW method, but neither performs as well as the linear predictor method used in [2]. The Electronics portion of this dataset was used for parameter tuning. These datasets were manually balanced such that each class represents 50% of the documents, which is not a very natural distribution.

In this case, we also chose to look at how a classifier trained and tested on all datasets together performed in comparison to the average of all classifiers. We found that both can train a classifier using all of the data that is slightly better than the average performance of the individual classifiers, however, training this classifier is very, very slow for the BOW method – so much slower we only performed 2x10 fold cross validation and used a different, faster, computer, rather than 10x10 cross validation as in the other cases, and it still took many times longer to train. On the other hand, the numeric method trains a combined classifier slightly faster than the sum of the individual numeric classifiers; these two methods clearly scale up very differently as we add more documents.

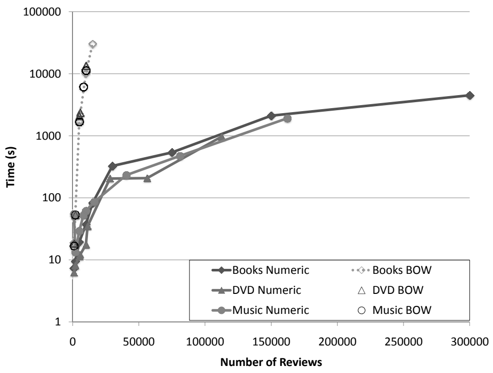


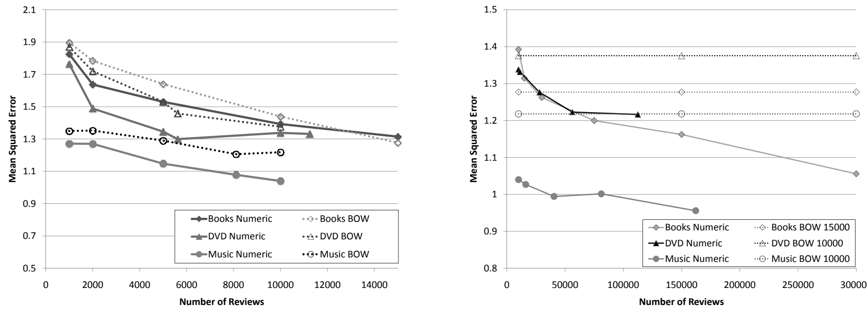
Fig. 2: Time required to train classifiers based on Numeric and Bag of Words features using varying amounts of data, 3 large Ordinal datasets

6.2 Very Large Datasets

This collection of data is a larger set of Amazon.com reviews from which the previous datasets were created. This collection included three domains with over 100,000 reviews – books, DVDs, and music, which allows us to explore how this approach scales to very large datasets. We used these larger datasets to examine how the numeric features scale in terms of both performance and time. In all cases, the results are reported on a single run using a 10,000 review test set (which is larger than most complete datasets used in previous research). These datasets are all highly imbalanced, with the majority class (5 star reviews) containing from 61-71% of the documents, and the minority class (2 star reviews) containing 4-6% of the reviews.

As shown in Figure 2, the time required to train with the numeric features scales much more gently than the time required to train with the BOW features. Note that the graph features a logarithmic scale for time. For time reasons, we only trained BOW based classifiers on up to 10-15,000 reviews, while we trained the classifiers using numeric features on over 100,000 reviews for each dataset, with 300,000 reviews for the Books dataset. For the books dataset, it took 8 hours and 23 minutes to train on 15000 documents with BOW features, while with the numeric features we were able to train on 300,000 documents in 1 hour and 15 minutes. In the case of the numeric dataset, the bulk of that time was to process the documents.

Performance in terms of MSE is shown in Figures 3a and 3b. Figure 3a shows performance of both approaches with up to 15,000 reviews. In this range, the numeric features are generally performing better by MSE, although at 15,000 Book reviews BOW is very slightly better. Figure 3b extends these performance results to show the space where we only tested the numeric features; the straight



(a) Up to 15,000 reviews

(b) Over 10,000 reviews

Fig. 3: Mean Squared Error, Numeric vs. BOW, 3 large Ordinal datasets

dotted lines represent the performance on the largest BOW classifier, 15,000 or 10,000 reviews depending on the dataset. This shows that if large amounts of documents are available, the numeric method continues to take advantage of them.

Similar results are obtained when looking at these datasets in terms of binary classification, with one and two star reviews as the negative class and four and five star reviews as the positive class.

6.3 Movie Review Datasets

We use a number of datasets created by Bo Pang & Lillian Lee in the domain of movie reviews. The movie review polarity dataset (version 2) and a dataset for sentence level subjectivity detection are introduced in [12], while a dataset for ordinal movie reviews by four different authors and a dataset for the sentiment of movie review "snippets" (extracts selected by RottenTomatoes.com) are introduced in [13].

Table 2 presents the results on the three binary datasets, as well as the results reported by Pang & Lee on the datasets, where available. While in the case of the Binary Movie reviews the numeric features fall well short of their reported results, on the Subjective Sentences dataset they are very close. Note that this dataset is for the related problem of subjectivity detection and not sentiment analysis. Pang & Lee report results on 10 fold cross validation, while we report results on 10 runs of 10 fold cross validation in order to be less sensitive to the random split of the data.

Table 3 reports the results on the ordinal movie reviews, by author. Again, we compare results to Pang & Lee, noting that we are approximating the values reported in a graph. Comparing to Pang & Lee based on accuracy, we find that in one case, the numeric features appear to be slightly better than their best result,

Table 2: Binary Datasets, Average Performance and Time, with 95% confidence intervals

	Accuracy	AUC	Time (m:ss)
Movie Review Polarity (Pang & Lee Accuracy: 0.872)			
Numeric	0.824 ± 0.005	0.896 ± 0.005	0:03
BOW	0.850 ± 0.005	0.850 ± 0.005	1:18
Movie Review Snippets			
Numeric	0.760 ± 0.002	0.841 ± 0.002	0:05
BOW	0.739 ± 0.002	0.739 ± 0.002	19:56
Subjective Sentences (Pang & Lee Accuracy: 0.92)			
Numeric	0.910 ± 0.002	0.967 ± 0.001	0:05
BOW	0.880 ± 0.002	0.880 ± 0.002	9:15

and in one other case, Pang & Lee's result is within the confidence range of our numeric features. In the two other cases, Pang & Lee's result is better than our result for the numeric feature set. However, we also note the comparison of their result to our simple BOW; in two cases our simple BOW classifier appears to be better, while in the other two the results are virtually the same. This confirms our assessment that this simple BOW is a good baseline to compare against.

Table 3: Ordinal Movie Reviews, BOW vs. Numeric, Accuracy, MSE, and Time, with 95% confidence intervals

Author	Type	MSE	Accuracy	Time (s)
Schwartz	Numeric	0.580 ± 0.013	0.518 ± 0.009	0.83
(0.51)[a]	BOW	0.691 ± 0.020	0.510 ± 0.009	13.20
Berardinelli	Numeric	0.478 ± 0.010	0.557 ± 0.008	1.48
(0.63)	BOW	0.443 ± 0.012	0.644 ± 0.007	35.64
Renshaw	Numeric	0.634 ± 0.016	0.468 ± 0.011	1.05
(0.50)	BOW	0.696 ± 0.020	0.496 ± 0.009	13.39
Rhodes	Numeric	0.490 ± 0.010	0.566 ± 0.007	1.65
(0.57)	BOW	0.478 ± 0.011	0.609 ± 0.006	43.79

[a] Accuracy obtained by Pang & Lee

7 Comparison with Feature Selection Methods

Another approach one might take to speeding up BOW is the idea of feature selection – selecting the most relevant features. In this section, we briefly compare the numeric features, plain BOW features, and BOW features reduced through two fast feature selection methods, Chi Squared and Information Gain. We use 5 fold cross validation on the binary electronics (2000 review balanced version),

Table 4: Feature Selection, Electronics

Method	Features	Accuracy	Time (s)
BOW	2855	0.786	19.25
Numeric	25	0.790	1.41
Chi Squared	100	0.789	5.70
Chi Squared	250	0.807	8.03
Chi Squared	1000	0.775	14.06
Information Gain	100	0.788	5.24
Information Gain	250	0.807	7.77
Information Gain	1000	0.780	13.52

subjective sentences, and movie review snippets datasets. These feature selection methods both evaluate individual attributes; methods which evaluate subsets of attributes together exist but are much slower [6].

On the electronics dataset, shown in Table 4, all classifiers complete in seconds but the numeric features are still the fastest. However, in this instance, the feature selection methods which select 250 features both achieve slightly higher accuracy than the numeric features, and, while slower, this difference in time may not be meaningful on a dataset of this size, as both complete in under 10 seconds.

Table 5: Feature Selection, Subjective Sentences

Method	Features	Accuracy	Time (m:ss)
BOW	4041	0.874	11:27.03
Numeric	25	0.911	0:04.24
Chi Squared	100	0.828	0:54.41
Chi Squared	250	0.860	1:12.81
Chi Squared	500	0.877	2:18.88
Chi Squared	1000	0.883	3:54.14
Information Gain	100	0.830	0:55.75
Information Gain	250	0.862	1:22.86
Information Gain	500	0.878	1:56.13
Information Gain	1000	0.883	3:03.52

As shown in Table 5, on the subjective sentences dataset, feature selection by both methods performed slightly better than plain BOW with 500 selected features and even better with 1000, and with substantial time savings over plain BOW. However, the numeric features are still much faster than any of the feature selection methods, and achieve the highest accuracy by a substantial margin.

Finally, on the Movie Review Snippets dataset, shown in Table 6, we again see that feature selection can save considerable time over plain BOW, and improve performance slightly with enough features, however, it is still orders of magnitude slower than the numeric features, and like on the subjective sentences dataset, the numeric method achieves the best performance.

Table 6: Feature Selection, Movie Review Snippets

Method	Features	Accuracy	Time (m:ss)
BOW	3688	0.732	37:21.3
Numeric	25	0.755	0:04.4
Chi Squared	100	0.655	0:58.5
Chi Squared	250	0.697	2:15.7
Chi Squared	1000	0.743	3:58.5
Chi Squared	1500	0.748	5:26.7
Information Gain	100	0.654	0:58.4
Information Gain	250	0.691	1:25.4
Information Gain	1000	0.743	4:24.9
Information Gain	1500	0.748	8:09.4

8 Discussion

This work decomposes the problem of learning the sentiment of documents into two simpler parts: scoring the strength of different words based on their distribution in positive and negative documents, and then learning document sentiment based on the distribution of those scores. This produces a compact representation, of around 25 features, compared with thousands for an effective BOW based approach.

While this decomposition results in the loss of some information – for instance, if two words appearing together in a document is significant – it appears as if the BOW representation may be too sparse for such relationships to be learned meaningfully. It seems as though the machine learning algorithms for the BOW representation are mainly learning which words are significant indicators of sentiment, but are much slower at this than simple word scoring methods. It should be noted that bag of words is, of course, a simplification of the original document as well, which loses information on word order.

This idea can be generalized to the concept of learning something about the attributes in a simple way and then using that knowledge to generate simpler features to which apply a more complicated learning approaches. In a slightly less general way, the numeric features group attributes together (in this case, through their scores as determined by the word scoring methods) and the machine learning algorithm learns based on the distribution of the attributes over the groups.

In the case of sentiment analysis, simple word scoring methods can learn the sentiment to a large extent, even when just computed as a simple threshold, but not as well as machine learning approaches.

While we have attempted to apply this feature extraction method to other non-text data sets with limited results, we believe that this exact approach would only work well in limited contexts outside of text data, where the attributes behave similarly to words, in that there is a large number of sparse attributes and the moderately infrequent ones are most useful.

The numeric features performed better than BOW in all respects on the two sentence level datasets. In addition, they also performed well on the online user reviews, both the smaller balanced datasets and the larger, imbalanced datasets. While the gap in performance narrowed on some datasets when comparing the largest trained BOW classifiers and the numeric classifiers trained with the same number of documents, the numeric features make training on very large datasets much more feasible, and we saw that performance continued to improve when using numeric features on larger and larger datasets.

On the document level movie review datasets, the results are mixed. With the ordinal datasets, the numeric features perform slightly better on accuracy on one of the authors, and we see the worst relative performance overall on two of the authors. However, we note that the MSE differences are relatively large in favor of the numeric features on two of the datasets, and relatively small in favor of BOW on the other two – on average, MSE prefers the numeric features. In the binary movie reviews, BOW has higher accuracy, and while the numeric features retain their advantage in terms of AUC, it is the smallest gap we see on that measure. These two datasets contain both relatively long and relatively well written material. While not conclusive, it may be that the numeric features are particularly adept at dealing with the shorter, less well written material found in all manner of less formal online discourse including online user reviews – which is a more interesting domain in many respects than professionally written reviews; there are far more bloggers than professional reviewers.

9 Conclusions

We have shown that it is possible to greatly condense the features used for machine learning of sentiment analysis and other related tasks for large speed improvements. Second, we have shown that these features often improve performance over a simple BOW representation, and are competitive with other published results. These speed improvements make it possible to process data sets orders of magnitude larger than previously attempted for sentiment analysis, which in turn generally leads to further performance improvements. This method is effective on both longer and shorter documents, as well as on small and large datasets, and may be more resilient to poorly written documents such as those found in online user reviews.

In addition, we have briefly compared this approach with a feature selection based approach. While feature selection can improve speed over plain BOW considerably, and can also increase performance, the numeric features remain considerably faster, particularly on larger datasets, and exceeded the performance of the best feature selection methods on two of the three datasets we examined, and were close on the other.

10 Future Work

We feel that this method has potential to be very useful as part of combination approaches to the problem, as a time saving replacement for the bag of words approach. We are also interested in developing the method to use a general vocabulary of scored words that can be trained on very large datasets, followed by customizing this method to domains with less data available.

Another avenue to explore involves the idea of partially combining the methods when dealing with very large datasets. It would be possible to compute word scores based on a subset of a large dataset and simply convert more training documents into that representation in order to train a machine learning algorithm, or to score the entire dataset and use a machine learning method on a smaller subset in order to speed up processing time even more for very large datasets.

References

1. A. Andreevskaia and S. Bergler. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, 2008.
2. J. Blitzer, M. Dredze, and F. Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, 2007.
3. K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinon extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, 2003.
4. L. Gaudette. Compact features for sentiment analysis. Master's thesis, University of Ottawa, 2009.
5. L. Gaudette and N. Japkowicz. Evaluation methods for ordinal classification. In *Proceedings of the twenty-second Canadian Conference in Artificial Intelligence (AI'2009)*, 2009.
6. M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1437–1447, November/December 2003.
7. A. Kennedy and D. Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 32(2):223–262, June 2006.
8. A. Lacey. A simple probabilistic approach to ranking documents by sentiment. In *Proceedings of the Class of 2005 Senior Conference*. Swarthmore College, 2005.
9. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. Massacheusetts Institute of Technology, 1999.
10. J. Martineau and T. Finin. Delta TFIDF: An improved feature space for sentiment analysis. In *Third AAAI Internatonal Conference on Weblogs and Social Media*, 2009.

11. T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.
12. B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271, Morristown, NJ, USA, 2004. Association for Computational Linguistics.
13. B. Pang and L. Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
14. B. Pang and L. Lee. *Opinion mining and sentiment analysis*, volume 2 of *Foundations and Trends in Information Retrieval.* Now, 2008.
15. F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*, 1998.
16. J. Wiebe, T. Wilson, and M. Bell. Identifying collocations for recognizing opinions. In *Proceedings of the ACL 01 Workshop on Collocation*, 2001.
17. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, San Francisco, 2nd edition, 2005.