Using multiple sources of agreement information for sentiment classification of political transcripts

Clint Burfoot
Department of Computer Science and Software Engineering
University of Melbourne, VIC 3010, Australia
c.burfoot@pgrad.unimelb.edu.au

Abstract

Sentiment classifiers attempt to determine whether a document expresses a generally positive or negative sentiment about its topic. Previous work has shown that overall performance can be improved by combining per-document classifications with information about agreement between documents. This paper explores approaches to sentiment classification of U.S Congressional floor debate transcripts that use a model for incorporating multiple sources of information about agreements between speakers. An empirical evaluation demonstrates accuracy improvements over previously published results.

1 Introduction

Thomas et al. (2006) investigate whether one can determine from the transcripts of U.S. Congressional floor debates whether speeches support or oppose a piece of proposed legislation. The authors exploit the fact that speeches occur as part of a discussion in order to incorporate information about their inter-relationships. Using this information provides substantial improvements over analysing speeches in isolation. The authors adopt a model, based on graph mincuts, that optimally balances classifications of speeches performed in isolation with information about whether or not pairs of speeches are likely to agree.

The work reported here mirrors Thomas et al. in using the same corpus and attempting the same binary sentiment classification task. It refines the the concept of agreement, comparing alternate techniques for detecting agreement in debates, and showing how these techniques can improve performance in the task of classifying speeches.

Two types of agreement information are introduced. The first, *party* agreement, takes advantage of the fact that speakers tend on average to vote with the majority of their party. A party-classifier is trained to detect whether a speaker is most likely a Republican or a Democrat. Links between pairs of speeches assigned to the same party encourage the overall classifier to give these speeches the same label. The second type of agreement is based on the intuition that speakers who agree often use similar words to describe their position. This *similarity* agreement is derived by comparing the context windows that surround references to the bill being considered in the debate.

An experimental evaluation shows that incremental accuracy improvements can be gained by making use of this agreement information. A range of technical limitations are discussed and ideas considered for future work that may make even better use of agreement information to improve sentiment classification.

2 Related Work

For a detailed survey of the field of sentiment analysis see (Pang and Lee, 2008).

The document-level sentiment-polarity classification task began as a “thumbs up or thumbs down” experiment. Pang et al. (2002) apply machine learning methods to predicting the overall sentiment of movie reviews. Turney uses PMI-IR to perform sentiment classification of reviews of banks, movies, cars and travel destinations (Turney, 2002; Turney, 2001).

Much work has focussed on the problem of separating language that contributes to an understanding of the sentiment of a document from language that is merely noise. Pang and Lee (2004) describe a sentence *subjectivity detector* that is trained on sets of labelled subjective and objective sentences. It embodies the intuition that important sentences will tend to cluster together by encouraging sentences to be classified as subjective or objective depending upon the classifications of their neighbours. Mullen and Collier (2004) isolate adjectives that occur in the same sentence as a reference to the topic of the document. Xia et al. (2008) use a semantic lexicon to extract sentiment words and surrounding words that serve to negate or modify the sentiment.

Work on document-level sentiment-polarity classification has built on attempts at determining *semantic orientation* of adjectives, i.e. whether an adjective indicates positive or negative sentiment. Kamps et al. (2004) grade adjectives on the bipolar adjective scales good/bad, active/passive and strong/weak using WordNet synonymy relationships. They allocate an adjective a score on each scale by finding the relative number of synonymy links that have to be traversed to get to the two pole adjectives. Hatzivassiloglou and McKeown (1997) extract sets of positive and negative adjectives from a large corpus using the insight that conjoined adjectives are generally of the same or different *semantic orientation* depending open the particular conjunction used. The results are specific to the domain of the corpus, which is helpful when adjectives have domain-specific orientations.

Some more recent sentiment classification papers have focussed on combining training data from multiple domains (Andreevskaia and Bergler, 2008; Li and Zong, 2008).

A number of studies have focussed on identifying agreement between speakers in the context of multi-party conversations. Galley et al. (2004), for example, describe a statistical model for identifying adjacency pairs and deciding if an utterance indicates agreement or disagreement with its pair. They classify using lexical (including *semantic orientation*), durational and structural features.

3 Corpus

Thomas et al. created the ConVote corpus using GovTrack[1], an independent website that collects publicly available data on the legislative and fund-raising activities of U.S. congresspeople. The HTML versions of GovTrack's floor-debate transcripts for 2005 were downloaded, cleaned, tokenised, and broken into debates and constituent speeches for use in the corpus.

Each speech is labelled with an identifier for its speaker and his recorded vote for the corresponding debate. Debates in which the losing side obtained less than 20% of the vote are omitted from the corpus on the grounds that they have less interesting discourse structure.

The debates are randomly allocated to training, development and test sets.

The text of each speech is parsed for references to other speakers (e.g. "I was pleased to work with the committee on the judiciary, and especially the gentleman from Virginia my good friend, to support the legislation on the floor today"), which are automatically tagged with the identity of the speaker being referred to.

An alternate version of the corpus is provided for use in classifying speeches in isolation. Agreement tags are not included. Short speeches that refer to yielding time (e.g. "Madame Speaker, I am pleased to yield 5 minutes to the gentleman from Massachusetts?") are removed on the grounds that they are purely procedural and too insubstantial for classification. Speeches containing the word "amendment" are removed because they tend to contain opinions on amendments rather than the bill being discussed.

4 Method

4.1 Classifying speeches using per-speech and inter-speech information

This work follows the method of Thomas et al. for incorporating information about inter-speech relationships into an overall classification. They draw on Blum and Chawla (2001), who describe how graph mincuts can balance per-element and pairwise information.

[1] http://govtrack.us

	total	train	test	development
speeches	3857	2740	860	257
debates	53	38	10	5
average number of speeches per debate	72.8	72.1	86.0	51.4
average number of speakers per debate	32.1	30.9	41.1	22.6

Table 1: Corpus statistics. Taken from Thomas et al. (2006)

Following Thomas et al., let $s_1, s_2, \ldots, s_n$ be the speeches in a given debate and let $\mathcal{Y}$ and $\mathcal{N}$ stand for the "supporting" and "opposing" classes, respectively. Assume a non-negative function *ind*(s, C) indicating a degree of preference for assigning speech s to class C. Also, assume some pairs of speeches have a link with non-negative strength, where str(ℓ) for a link ℓ indicates a degree of preference for the linked speeches to be assigned to the same class. The class allocation $c(s_1), c(s_2), \ldots, c(s_n)$ is assigned a cost

$$\begin{aligned} c \;=\; & \sum_s \mathit{ind}(s, \bar{c}(s)) \qquad (1) \\ & + \sum_{s,s'\colon\, c(s)\neq c(s')} \; \sum_{\ell \text{ between } s,s'} \mathrm{str}(\ell) \end{aligned}$$

where $\bar{c}(s)$ is the complement class to $c(s)$. A *minimum-cost* assignment then represents an optimum way to balance a tendency for speeches to be classified according to their individual characteristics with a tendency for certain speech pairs to be classified the same way. The optimisation problem can be solved efficiently using standard methods for finding minimum cuts in graphs.

4.2 Classifying speeches in isolation

Whole-of-speech classification: Pang et al (2002) show that standard machine-learning methods can be used to classify documents by overall sentiment, with best results coming from support vector machines (SVM) with unigram presence features for all words. Punctuation characters are retained and treated as separate tokens and no stemming is done. Their approach is adopted to classify speeches in isolation using the popular SVM^{light} [2] classifier with default parameters (Joachims, 1999).

Following Thomas et al. speeches are allocated an *ind* value based on the signed distance $d(s)$ to the SVM decision plane:

$$\mathit{ind}(s, \mathcal{Y}) \overset{\text{def}}{=} \begin{cases} 1 & d(s) > 2\sigma_s; \\ \left(1 + \frac{d(s)}{2\sigma_s}\right)/2 & |d(s)| \leq 2\sigma_s; \\ 0 & d(s) < -2\sigma_s \end{cases}$$

where σ_s is the standard deviation of $d(s)$ over all of the speeches s in the debate and *ind*$(s, \mathcal{N}) = 1 -$ *ind*$(s, \mathcal{Y})$.

4.3 Classifying agreements between speeches

Same-speaker agreements: Speakers in congressional floor-debates will often contribute more than one speech. As Thomas et al. note, one can imagine that if a political debate is serving its purpose a speaker might change his mind during its course, so that one of his later speeches contradicts an earlier one. Unfortunately, this scenario has to be ruled of consideration since our corpus labels speeches based on the speaker's final vote. To represent this simplification, each of the speeches by a given speaker is linked to another with a link ℓ of infinite strength. This guarantees that they will receive the same final classification [3].

Reference agreements: Speakers in congressional floor-debates sometimes refer to each other by name. These references are labelled in the corpus. Thomas et al. build an agreement classifier to take advantage of the intuition that the words a speaker uses when referring to another speaker will give a clue as to whether the two agree. They use an SVM classifier trained on the tokens immediately surrounding[4] the reference. Following Thomas et al.

[2] http://svmlight.joachims.org/

[3] Detecting the point at which speakers change their minds could make an interesting area for further research.

[4] The authors find good development set performance using the window starting 30 tokens before the reference and ending 20 tokens after it.

let $d(r)$ denote the distance from the vector representing the reference r to the SVM decision plane and let σ_r be the standard deviation of $d(r)$ over all references in the debate. Define the strength *str* of the link as:

$$\mathit{str}(r) \stackrel{\text{def}}{=} \begin{cases} 0 & d(r) < \theta_{\text{agr}}; \\ \alpha_{\text{r}} \cdot d(r)/4\sigma_r & \theta_{\text{agr}} \leq d(r) \leq 4\sigma_r; \\ \alpha_{\text{r}} & d(r) > 4\sigma_r \end{cases}$$

where θ_{agr} is a threshold that increases the precision of the links by discarding references that are not classified with enough confidence and α_{r} represents the relative strength of reference links.

Same-party agreements: A brief examination of the ConVote corpus confirms the intuition that speakers tend to vote with their party. This is a form of agreement information. If we know that two parties "agree" on their choice of party affiliation, we can conclude they are more likely to vote the same way on any given bill. The method already described for whole-of-speech classification can be trivially extended for party detection by substituting party labels for vote labels.

It is reasonable to assume that by-name references to other speakers also give a hint about party affiliation. A reference classifier is trained with party labels, using the method described in the section above.

An overall party classification, $p(s)$, is derived for the whole-of-speech and reference agreement classifiers using the graph mincut method already described. Define the strength $\mathit{pstr}(s, s')$ of the agreement link as:

$$\mathit{pstr}(s, s') \stackrel{\text{def}}{=} \begin{cases} \alpha_{\text{p}} & p(s) = p(s'); \\ 0 & p(s) \neq p(s') \end{cases}$$

where α_{p} represents the relative strength of party links.

Similarity agreements: To measure the extent to which a pair of speakers use similar words to describe their positions, let $\mathit{sim}(s, s')$ be the similarity of two speeches in a debate determined with the standard information retrieval measure of cosine similarity with tf.idf term weighting (Manning et al., 2008). Define a link strength $\mathit{bstr}(s, s')$ as:

$$\mathit{bstr}(s, s') \stackrel{\text{def}}{=} \begin{cases} \mathit{sim}(s, s') \cdot \alpha_{\text{b}} & \mathit{sim}(s, s') > 4\sigma_{\text{b}}; \\ 0 & \mathit{sim}(s, s') \leq 4\sigma_{\text{b}} \end{cases}$$

where σ_{b} is the standard deviation of $\mathit{sim}(s, s')$ over all of the speeches in the debate and α_{b} represents the relative strength of similarity links. The use of the threshold based on standard deviation serves to limit the links to the most strongly similar pairs without introducing another free parameter.

To reduce noise, the input to the similarity algorithm is limited to the set of tokens that appear within a fixed window of the tokens "bill" or "h.r."[5]. These two tokens tend to indicate that the speaker is commenting directly on the bill that is the topic of the debate.

Overall classification: To incorporate these two new measures of agreement into the overall classification framework, assign a new value for *c* by modifying equation (1) as

$$\begin{aligned} c \;=\; & \sum_{s} \text{ind}(s, \bar{c}(s)) \qquad (2) \\ & + \sum_{s,s' \colon c(s) \neq c(s')} \left\{ \begin{array}{rl} & \mathit{pstr}(s, s') \\ + & \mathit{bstr}(s, s') \\ + & \displaystyle\sum_{\ell \text{ between } s,s'} \text{str}(\ell) \end{array} \right\} \end{aligned}$$

4.4 A note on relevance

Impressionistically, a significant proportion of speeches are not clearly relevant to the bill. Even with the "amendment" and "yield" speeches removed, there are a variety of procedural utterances and instances where speakers are evidently dealing with some unrelated bill or motion. These spurious speeches are problematic as they skew the final accuracy figures and may reduce classifier accuracy by introducing noise. An early iteration of this work used official summaries of the bills to build a relevance metric. Speeches that had high tf.idf similarity with the bill summary were considered more relevant. This figure was used to test three hypotheses.

1. Speeches that have a higher relevance will be more accurately classified by the whole-of-speech classifier.

[5] Good development set performance is obtained using the window starting 15 tokens before the reference and ending 15 tokens after it.

Similarity agreement classifier (“similarity⇒agreement?”)	Devel. set	Test set
majority baseline	49.02	49.02
classifier	61.09	59.94

Table 2: Similarity agreement accuracy, in percent.

2. Reference agreements between pairs of speeches that have higher relevance will be classified more accurately.

3. Similarity agreements between pairs of speeches that have higher relevance will be more accurate.

Early experiments did not support these hypotheses, so the approach was abandoned. Nevertheless a more sophisticated measure of relevance may eventually prove to be a key to improved accuracy.

5 Evaluation

This section presents experiments intended to evaluate the performance of the classifier against baselines and benchmarks. Ten-fold cross validation is used for all but one experiment, with 8 parts of the data designated for training, 1 for testing and 1 for development. The development set is used for tuning free parameters. The tuning process consists of repeatedly running the experiment with different values for the free parameters α_{r}, α_{p} and α_{b}. The combination of values that gives the best result is then used for final evaluation against the test set.

The last experiment is a comparison with the results from Thomas et al.

5.1 Similarity agreements

The baseline for similarity agreement classification is the percentage of possible speech pairs that agree. This is approximately half. Table 2 shows that the classifier predicts agreement with about 60% accuracy.

5.2 Reference classification

The baseline for reference classification is the percentage of by-name references that correspond with agreement across the whole corpus. The relatively high figure is evidence that speakers tend to refer to the names of others with whom they agree. The value for θ_{agr} is not optimised. Just two values were tried: 0 and μ, the average decision-place distance across all non-negative scores. As shown in Tables 3 and 4, the use of a non-zero cutoff introduces a precision-recall tradeoff.

Agreement classifier (“reference⇒agreement?”)	Devel. set	Test set
majority baseline	81.48	80.23
$\theta_{\mathrm{agr}} = 0$	80.39	80.80
$\theta_{\mathrm{agr}} = \mu$	89.59	89.48

Table 3: Agreement-classifier accuracy, in percent. References that do not meet the threshold are not counted.

Same-party classifier (“reference⇒same-party?”)	Devel. set	Test set
majority baseline	77.16	79.68
$\theta_{\mathrm{agr}} = 0$	81.31	76.23
$\theta_{\mathrm{agr}} = \mu$	81.44	78.74

Table 4: Same-party-classifier accuracy, in percent. References that do not meet the threshold are not counted.

An early version of this experiment attempted to infer agreements from disagreements. Two speakers who disagree with a third speaker must, by definition, agree with each other, since speakers can only vote to support or oppose. Two factors limited the usefulness of this approach. First, less than 20% of references correspond with disagreement. Speakers seem to prefer referring by name to others with whom they agree. Second, the agreement classifier did not do a reliable job of detecting these.

5.3 Party classification

An audit of the corpus shows that, averaged across all debates, 92% of votes concur with the party majority. This should mean that the 85% accurate labels obtained by the party classifier, shown in Table 5 should make prediction of votes significantly easier.

An alternative to party classification would have been to take the party labels as input to the vote classifier. After all, it is reasonable to assume that any real-world application of sentiment classification of formal political debate could rely on knowledge of the party affiliation of the speakers. Two factors make it more interesting to limit the use of prior

Republican/Democrat classifier (“speech⇒Republican?”)	Devel. set	Test set
majority baseline	51.08	51.08
SVM [speech]	68.97	69.35
SVM + same-speaker-links ...		
+ agreement links; $\theta_{\text{agr}} = 0$	81.31	76.23
SVM + same-speaker-links ...		
+ agreement links; $\theta_{\text{agr}} = \mu$	85.22	84.26

Table 5: Republican/Democrat-classifier accuracy, in percent.

Support/oppose classifier (“speech⇒support?”)	Devel. set	Test set
majority baseline	53.85	53.85
SVM + same-speaker-links ...		
+ agreement links, $\theta_{\text{agr}} = \mu$	81.77	79.67
+ agreement links, $\theta_{\text{agr}} = \mu$		
+ party links	85.07	80.50
+ agreement links, $\theta_{\text{agr}} = \mu$		
+ similarity links	81.77	79.67

Table 6: Support/oppose classifier accuracy, in percent.

knowledge for the purposes of this paper. First, there are cases where party affiliation will not be available. For example, it is helpful to know when independent speakers in a debate are using language that is closer to one party or the other, since their vote on that debate will probably tend accordingly. Second, this approach better demonstrates the validity of the classification model for use with imperfect domain knowledge. Such techniques are more likely to be applicable in informal domains such as analysis of political text in the blogosphere.

5.4 Overall classification

The benchmark for evaluating the utility of party and similarity agreements is the score for in-isolation classification combined with same speaker links and reference agreements. This is represented in Table 6 as “SVM + same-speaker-links + agreement links, $\theta_{\text{agr}} = \mu$”. Adding in party links improves accuracy by about 1% on the test set and 3% on the development set.

Adding similarity links does not improve overall accuracy. It is somewhat surprising that party links do better than similarity links. Similarity links have the advantage of existing in variable strengths, depending upon the degree of similarity of the two speeches. It may be that the links are simply not reliable enough. It seems likely that the three relatively simple methods used in this work for detecting agreement could be improved with further research, with a corresponding improvement to overall classification accuracy.

5.5 Comparison with previous results

Thomas et al. obtained a best result of 70.81% with $\theta_{\text{agr}} = 0$ and no cross-validation. The equivalent best result produced for this experiment using the same algorithm was 71.16%, a difference probably due to some minor divergence in implementation. The addition of party links gives no accuracy increase. Adding similarity links gives an accuracy increase of about 0.5% on both the development and test sets.

As shown in Table 7, the development set results for the Thomas et al. experiment are much better than those obtained using cross-validation. The test set results are much worse. This is surprising. It appears that the Thomas et al. split is unlucky in the sense that differences between the development and test sets cause unhelpful tuning. The use of cross validation helps to even out results, but the arbitrary split into sets still represents a methodological problem. By using 8 sets of debates for training, 1 for testing and 1 for tuning, we are open to luck in the ordering of sets because the difference between good and poor results may come down to how well the development set that is chosen for tuning in each fold happens to suit the test set. One solution would be to cross-validate for every possible combination of test, development and training sets.

5.6 Choice of evaluation metric

Since our choice of ground truth precludes the speeches by a speaker in a single debate from having different labels, the decision to evaluate in terms of percentage of speeches correctly classified seems slightly questionable. The alternative would be to report the percentage of speakers whose votes are

Support/oppose classifier (“speech⇒support?”)	Devel. set	Test set
majority baseline	54.09	58.37
SVM + same-speaker-links ...		
+ agreement links, $\theta_{\text{agr}} = \mu$	89.11	71.16
+ agreement links, $\theta_{\text{agr}} = \mu$ + party links	89.11	71.16
+ agreement links, $\theta_{\text{agr}} = \mu$ + similarity links	89.88	71.74

Table 7: Support/oppose classifier accuracy, in percent, using the training, development, test debate split from Thomas et al. without cross-validation.

correctly predicted. This approach more correctly expresses the difficulty of the task in terms of the number of degrees of freedom available to the classifier, but fails to consider the greater importance of correctly classifying speakers who contribute more to the debate. This work has retained the established approach to allow comparison. More comprehensive future works might benefit from including both measures.

6 Conclusions and future work

This study has demonstrated a simple method for using multiple sources of agreement information to assist sentiment classification. The method exploits moderately reliable information about whether or not documents agree in order to improve overall classification. Accuracy suffers somewhat because of the need to tune link strengths on a set of development data. Future work should attempt to remove this limitation by developing a more principled approach to incorporating disparate information sources.

There is great scope for exploration of the concept of agreement in sentiment analysis in general. Being able to detect whether or not two documents agree is likely to be useful in areas beyond sentiment classification. For example, tools could assist researchers in understanding the nuances of contentious issues on the web by highlighting areas in which different sites or pages agree and disagree. eRulemaking tools that cross-link and group public submissions about proposed legislation could benefit from being able to match like opinions. Matching could be on the basis of overall opinion or a breakdown of the issue into separate aspects. Sentiment summarisation tools could be aided by the ability to group content from separate documents into sections that have been judged to have equivalent sentiment.

Document-level sentiment classification techniques are limited by their inability to reliably ascribe expressions of sentiment. Strongly positive or negative expressions may relate to an aspect of the topic that is linked to overall sentiment in an unpredictable way. For example, a speaker offers the following to a debate on establishing a committee to investigate the preparation for, and response to, Hurricane Katrina: “Mr. Speaker, the human suffering and physical damage wrought by Hurricane Katrina is heart-wrenching and overwhelming. We all know that very well. Lives have been lost and uprooted. Families are separated without homes and without jobs.” This extract is full of strongly negative words. Nevertheless, it comes from a speech given in support of the bill. It is unlikely that a bag-of-words polarity classifier will be able to separate the negative sentiment expressed about Hurricane Katrina from any sentiment that is expressed about the bill itself.

As another example, the following quotes are from two contributors to a debate on protected species legislation. “The E.S.A has only helped 10 of 1,300 species listed under the law. Thirty-nine percent of the species are unknown. Twenty-one percent are declining, and they are declining, and 3 percent are extinct. This law has a 99 percent failure rate.” “Mr. Chairman, the endangered species act is a well-intentioned law that has failed in its implementation. Originally billed as a way to recover and rehabilitate endangered species, it has failed at that goal.” Both of these quotes use apparently negative language, yet they are given in support of the proposed legislation. The negative opinion being expressed is about legislation which is to be replaced by the bill under debate.

One way for a classifier to deal with this kind of material is to make the rhetorical connection between negative sentiment about an existing bill and support for its replacement. This is quite a challenge. An alternative is to make use of agreement links. If the classifier can successfully detect that the negative sentiment in both speeches relates to the "E.S.A." it can correctly determine that the two agree on at least part of the issue. This information can then be combined with a judgement from a per-document classifier that is trained to consider only parts of speeches that refer directly to the bill being discussed. A three-stage process might be: (i) Performing in-isolation classification based on a document extract that is deemed to express overall sentiment; (ii) Performing agreement classification on document extracts that are deemed to relate to distinct aspects of the debate; and (iii) Completing an overall classification based on these per-document and inter-document measures. Future work could focus on developing this approach.

References

Alina Andreevskaia and Sabine Bergler. 2008. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of ACL-08: HLT*, pages 290–298, Columbus, Ohio, June. Association for Computational Linguistics.

Avrim Blum and Shuchi Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, pages 19–26. Morgan Kaufmann, San Francisco, CA.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 669–676, Barcelona, Spain, July.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain, July. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

J. Kamps, M. Marx, R. Mokken, and M. de Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, volume IV, pages 1115–1118.

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *Proceedings of ACL-08: HLT, Short Papers*, pages 257–260, Columbus, Ohio, June. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 412–418, Barcelona, Spain, July. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 271–278, Barcelona, Spain, July.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundation and Trends in Information Retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 Conference on Empirical methods in Natural Language Processing*, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia, July. Association for Computational Linguistics.

Peter D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *ECML '01: Proceedings of the 12th European Conference on Machine Learning*, pages 491–502, London, UK. Springer-Verlag.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Yunqing Xia, Linlin Wang, Kam-Fai Wong, and Mingxing Xu. 2008. Lyric-based song sentiment classification with sentiment vector space model. In *Proceedings of ACL-08: HLT, Short Papers*, pages 133–136, Columbus, Ohio, June. Association for Computational Linguistics.