AUTOMATIC ANALYSIS OF DOCUMENT SENTIMENT

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Bo Pang

August 2006

AUTOMATIC ANALYSIS OF DOCUMENT SENTIMENT

Bo Pang, Ph.D.
Cornell University 2006

Sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in text, has attracted a great deal of attention. Potential applications include question-answering systems that address opinions as opposed to facts and business intelligence systems that analyze user feedback. The research issues raised by such applications are often quite challenging compared to fact-based analysis. This thesis presents several sentiment analysis tasks to illustrate the new challenges and opportunities. In particular, we describe how we modeled different types of relations in approaching several sentiment analysis problems; our models can have implications outside this area as well.

One task is polarity classification, where we classify a movie review as "thumbs up" or "thumbs down" from textual information alone. We consider a number of approaches, including one that applies text categorization techniques to just the subjective portions of the document. Extracting these portions can be a hard problem in itself; we describe an approach based on efficient techniques for finding minimum cuts in graphs that incorporate sentence-level relations. The second task, which can be viewed as a non-standard multi-class classification task, is the rating-inference problem, where one must determine the reviewer's evaluation with respect to a multi-point scale (e.g. one to five "stars"). We apply a meta-algorithm, based on a metric-labeling formulation of the problem, that explicitly exploits relations between classes. A different type of relationship between text units is considered in the third task, where we investigate whether one can determine from

the transcripts of U.S. Congressional floor debates whether the speeches represent support of or opposition to proposed legislation. In particular, we exploit the fact that these speeches occur as part of a discussion. We find that the incorporation of information regarding relationships between discourse segments yields substantial improvements over classifying speeches in isolation. Lastly, we introduce and discuss a sentiment analysis problem arising in Web-search applications: given documents on the same focused topic, we wish to rank the subjective documents before objective ones. We present early results with unsupervised approaches that do not assume prior linguistic or domain-specific knowledge.

BIOGRAPHICAL SKETCH

Bo Pang received her B.S. degree in Computer Science from Tsinghua University. She is currently a Ph.D. student in Computer Science at Cornell University.

ACKNOWLEDGEMENTS

I am eternally grateful to Lillian Lee, who has been an amazing advisor in so many aspects of this role. Apart from numerous other things (both research- and non-research-wise) that I have learned from her, it was her introduction to statistical methods and models that got me interested in NLP in the first place. Thanks to her support, patience, sense of humor, and consistently wise advice, I have greatly enjoyed working in this field. On top of all these, what touches me the most is that she truly cares[1]; for this alone I can never thank her enough. Shimon Edelman, my minor advisor, has been a great resource. I learned much of what I know about cognitive science from him; and during bad times in my research, attending his lectures and seminars restored in me a sense of being intellectually alive. Many thanks also to the other members on my committee — Claire Cardie, Thorsten Joachims, and Eva Tardos, for many helpful discussions and feedback, as well as constant encouragement and valuable advice.

The work described in this thesis undoubtedly benefited from discussions with my committee members; in addition, I would like to also thank Paul Bennett, Dave Blei, Eric Breck, Rich Caruana, Yejin Choi, Dan Huttenlocher, Jon Kleinberg, Oren Kurland, John Lafferty, Guy Lebanon, Michael Macy, Art Munson, Andrew Myers, Vincent Ng, Fernando Pereira, Pradeep Ravikumar, Phoebe Sengers, Ves Stoyanov, David Strang, Richard Tong, Peter Turney, Ramin Zabih, Jerry Zhu, and the anonymous reviewers for many valuable comments and helpful suggestions.

Portions of this work were started while I was visiting IBM Almaden as a summer intern. I really appreciate all the help and input I received from my mentor Shivakumar Vaithyanathan. ISI is undoubtedly a wonderful place to do

[1] I hope she won't be rolling her eyes at this with "what is this? 'vote for Lillian; she cares?' " — although quite likely she will.

NLP research. Many thanks to my mentors, Kevin Knight and Daniel Marcus, for their insights and advice. I also greatly enjoyed my interactions with the NLP group there, and especially want to thank Hal Daumé III, Ulrich Germann, Ulf Hermjakob, and Radu Soricut for help and discussions. For yet another happy summer in California and valuable experience with the "real world", I thank my host at Google, Xuefu Wang. I also thank CMU for their hospitality during my one-year visit; it has been an extremely enriching experience for me.

The computer science department in Cornell has been a most supportive and friendly community. It is indeed an impossible task to name all the people that have helped and influenced me over the years and to thank them individually. I am especially thankful to the NLP group and the machine learning people. I have always enjoyed both academic and casual conversations at the NLP reading group, AI lunch, and MLDG (thanks to Eric Breck, Alexandru Niculescu-Mizil, and Filip Radlinski for putting so much thought into it; you can now, once again, enjoy being the most senior students!); as well as Friday beers at CTB and Big Red Barn. I also enjoyed many inspiring conversations with Regina Barzilay when she visited Cornell. Her enthusiasm has always put a positive spin on me. Another positive force was our resourceful Cindy Robinson; many thanks to her for her help, laughters, and candies. I have been blessed with many great officemates throughout the years. I thank Greg Bronevetsky, Hubie Chen, Tony Faradjian, Dan Kifer, Oren Kurland, Xiangyang Lan, and Yong Yao (and Elliot Anshelevich and Leonid Meyerguz, whose visits to 4107 were at least partially responsible for our noise level!) for making the graduate school years so much easier.

Life would have been much different without the "dinner gang" — Hailing, Lin, Tina, Yanling, Min and Min at the "Ithaca headquarter", as well as Ting and

Minglong, when I was in Pittsburgh. Thanks for all the happy dinners and countless indecisive hours we wasted together; thanks also for tolerating my occasional outburst of "artistic impulse" and my not so occasional urge to argue.

Last but not least, deepest thanks to my parents, who have always stood behind me for who I am. Their trust and support have kept me grounded, and made me believe in myself when I couldn't find any reasons to.

This thesis is based upon work supported in part by the National Science Foundation under grant no. IIS-0081334, IIS-0329064, and CCR-0122581; SRI International under subcontract no. 03-000211 on their project funded by the Department of the Interior's National Business Center; a Cornell Graduate Fellowship in Cognitive Studies; Google Anita Borg Memorial Scholarship funds; and by an Alfred P. Sloan Research Fellowship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

> *Will, you must forgive me, but I have not the slightest sympathy with what the world calls Sentiment – not the slightest.*
>
> — Mark Twain, *Letter to Will Bowen*

A large portion of research in natural language processing seeks to better understand and process various types of information in text. In the past, the majority of such work focused on factual information: work in text categorization largely concentrated on classifying documents according to their subject matter (e.g., politics vs. sports); question answering systems aimed to answer fact-oriented questions such as "who did what, when and where"; similarly, research in text summarization mainly focused on collections of news articles where it is important to "keep the facts straight". Indeed, text segments such as "Cornell is located in Ithaca, NY" contain nothing but facts. But if we look at the examples shown in Figure 1.1, although it is useful to detect factual information such as both text segments are *about* Cornell University, it is just as useful (and perhaps even more interesting) to understand the author's *sentiment* indicated by the text.

Quoting the first two senses given by the Merriam-Webster Online Dictionary,

Excerpt of Review 1 (by pandora)

School, Friends, Sleep. Choose 2.

Mar 27 '00

Pros *Beautiful campus*

Cons *everything else*

Ahhh? Fair Cornell University? High above Cayuga's waters.

And pretty much that is all that is good about Cornell University.

Excerpt of Review 2 (by heafdog)

I would cut off a finger to do it all again.

Aug 16 '01

Pros *Beautiful campus, excellent size, terrific experience.*

Cons *Can't think of any.*

When I say that I would cut off a finger to do it all over again, it sounds sort of drastic, but I really mean that. I can go on and on for hours about how much I loved Cornell. It was the best four years of my life, hands down. In the evening after my graduation from there, back in 1998, I was so sad to be leaving there that I cried my eyes out. Seriously.

Figure 1.1: Reviews on Cornell University from Epinions.com.

the word *sentiment* can be used to indicate

1. **a**: an attitude, thought, or judgment prompted by feeling : PREDILECTION

   b: a specific view or notion : OPINION

2. **a**: EMOTION **b**: refined feeling : delicate sensibility especially as expressed in a work of art **c**: emotional idealism **d**: a romantic or nostalgic feeling verging on sentimentality

It is the first sense ("attitude, judgment, and view") of sentiment, rather than the "delicate feeling" or "nostalgic feeling verging on sentimentality"[1], that we attempt to address in this work. In particular, we focus on the computational analysis of such information in text.

For instance, one salient feature of the texts shown in Figure 1.1 is the *sentiment polarity*. The first review clearly indicates a negative opinion towards Cornell, whereas the second one, in spite of associating Cornell with the (presumably negative) image of cutting off a finger, conveys positive sentiment. While it is fairly easy for human readers to determine the sentiment polarity of these two examples, it is quite a challenge for machines to accomplish the same task automatically. As we know, most text classification methods (i.e., techniques that classify a given document into one of the pre-defined classes) are based on "bag of words" models, where each document is represented by the set of words it contains, ignoring the original order in which these words appear. Simple as it is, looking at the statistics of the constituent words alone has proven quite effective for topic-based text classification — after all, it is quite reasonable to assume that texts about Cornell are more likely to contain certain keywords (such as, obviously, "Cornell", or, not so obviously, "Cayuga"). In contrast, in the case of sentiment-based classification, while the positive review in Figure 1.1 does contain "positive" keywords such as "loved" and "best", the negative review contains no apparent "negative" words. In fact, plenty of "positive" words such as "beautiful" and "fair" are used in this

[1]It seems it is this second sense of "sentiment" that Mark Twain does not have sympathy with.

negative example. In general, while topics (e.g., politics or sports) are often identifiable by keywords alone, sentiment can be expressed in a more subtle manner and seems to require deeper analysis. We will discuss this particular problem in more detail in Chapter 3. It suffices to say that compared to traditional fact-based analysis, sentiment-oriented text analysis problems bear unique characteristics and present us with interesting new challenges.

Indeed, sentiment analysis, or the computational treatment of opinion, sentiment, and subjectivity, has recently attracted a great deal of attention (see Chapter 2 for a survey on related work). In addition to the research challenges, the growing interest in this area is also partly due to the many potential applications. The texts shown in Figure 1.1 are not two isolated examples dug out from a singular site dedicated to online reviews. Recent years have seen rapid growth in various kinds of on-line discussion groups and review sites, such as the *New York Times* Books web page or the Internet Movie Database (IMDb), where a crucial characteristic of the messages posted is their sentiment, or overall opinion towards the subject matter (e.g., whether a product review is positive or negative, as we just discussed above). Labeling such messages with their sentiment would provide succinct summaries to readers; indeed, these labels are part of the appeal and value-add of sites like `www.rottentomatoes.com`, which both labels movie reviews that do not contain explicit rating indicators and normalizes the different rating schemes of many reviewers. Similarly, many other interesting applications can benefit from automatic analysis of sentiment in verbal contexts: business intelligence applications can use summaries of user feedback; recommender systems (e.g. Terveen et al. (1997)) may attain better user profiles by analyzing textual comments; information-extraction and question-answering systems can be enhanced to

deal with statements and queries regarding opinions rather than facts (Cardie et al., 2003). Potential applications also include message filtering, in that perhaps one could use sentiment information to recognize and discard "flames"(Spertus, 1997).

In addition to the interesting characteristics of sentiment analysis problems, we also investigate the modeling of relations in text collections. The data commonly used in natural language processing (NLP) research in the past tended to be clean, edited, and grammatical, as they were dominated by collections of text published by traditional media in the form of books or newspapers. Nowadays, much of the textual data available on the web, such as blog postings or user reviews, tends to be less structured and sometimes even less well-formed, containing incorrect spellings or grammar mistakes. While such "flawed" data may be more resistant to traditional NLP systems such as full parsers, there is still interesting "beyond-bag-of-words" information to explore. In particular, various types of relations exist at different levels of text collections. For instance, at the sentence level, proximity can be an interesting relation to model since nearby sentences in a document may share certain properties we are interested in. One focus of this work is the mathematical modeling of such relations within machine learning approaches. We believe the techniques we explored are not unique to sentiment analysis and should have broader implications.

The majority of the work described in this thesis focuses on classification tasks. Most of the problems are investigated in the context of classifying a given opinion-bearing document (e.g., a review or the transcript of a political speech) according to the sentiment polarity (a two-class problem) or degree of positivity (a multi-class problem) expressed in the text. A form of sentiment summarization is consid-

ered in this context by extracting the subjective portions of an input document, touching upon the problem of subjectivity detection We also examine a ranking problem with the goal of identifying opinion-bearing documents in a web-search setting, where we experimented with completely unsupervised approaches. Thus, this thesis is organized as follows. Chapter 2 gives a brief survey on related work; Chapters 3, 4, and 5 consider several sentiment-oriented classification problems, with a particular emphasis on modeling relations; Chapter 6 describes our preliminary study of the ranking problem in the sentiment context; and we conclude with Chapter 7.

Contributions: This thesis describes explorations of several sentiment-analysis problems from a purely data-driven perspective. We do believe deeper linguistic knowledge can further improve the performance, and it is important to note that many methods described in this thesis can easily incorporate prior knowledge, complementing other work with more focus on feature engineering or linguistic knowledge. But our main focus in this work was to explore how much can be learned from the raw data. In fact, it can be surprising how much machine learning approaches can achieve on seemingly sophisticated problems like sentiment polarity classification.

Whenever possible, we tried to carry out pilot studies to take closer looks at the problem being considered before exploring more sophisticated methods. This way, we gain a better understanding of the actual difficulty of the problem in our specific setting, rather than trusting our intuition, or prior belief.

This thesis also examines non-labor-intensive ways to collect interesting datasets. The corpus we constructed range from collections of movie reviews to transcripts

political debates, with labels extracted automatically from online resources. These datasets enable as well as motivate the problems explored in this thesis, serving as the basis for our experiments with machine-learning methods.

We show that incorporating various types of relationship information through conceptually simple, yet flexible frameworks can provide significant improvements over relation-blind methods. In particular, we describe an approach based on efficient techniques for finding minimum cuts in graphs to incorporate sentence-level relations within a given document as well as document-level relations between speeches that occur as part of a discussion. We also consider a meta-algorithm, based on a metric-labeling formulation, that explicitly exploits relations between classes for a multi-class classification problem.

Chapter 2

Related work

There has recently been a dramatic surge of interest in sentiment analysis, as more and more people become aware of the scientific challenges posed and the scope of new applications enabled by the processing of subjective language. The papers collected by Qu, Shanahan, and Wiebe (2004) form a relatively early representative sample of research in the area; and Esuli (2006) maintains an active bibliography. Section 2.1 briefly surveys a range of different sentiment analysis tasks that formed the "environment" within which the work described in this thesis was developed. Section 2.2 discusses other work that investigated non-factual information in text. We leave the discussion of related work that is less focused on sentiment to later chapters as it is more relevant to other aspects of the specific problems we consider in this work.

2.1 Related work in sentiment analysis

Polarity classification As mentioned earlier, the task of *sentiment polarity classification*, where we want to classify an opinionated document as either positive or negative according to the sentiment expressed by the author, has attracted a lot

of interest in past years. (Note that the rating-inference problem, where one must determine the reviewer's evaluation with respect to a multi-point scale (e.g. one to five "stars") can be viewed as a multi-class version of this problem. See Chapter 4 for related work specific to that problem.)

Early work on sentiment-based categorization of entire documents has often involved either the use of models inspired by cognitive linguistics (Hearst, 1992; Sack, 1994) or the manual or semi-manual construction of discriminant-word lexicons (Huettner and Subasic, 2000; Das and Chen, 2001; Tong, 2001). For instance, Das and Chen (2001) presented a methodology for real time sentiment extraction in the domain of finance; working with messages from web-based stock message boards, attempt to automatically label each such message as a "buy", "sell" or "neutral" recommendation. Their classifier achieves an accuracy of 62% (the upper bound, human agreement rate, was 72%). Unfortunately, their method entails constructing a discriminant-word lexicon by manual selection and tagging of words from several thousand messages. This is something we wish to avoid in our work.

In contrast, with (automatically) labeled data collected from the web, we approached the related task of determining sentiment polarity in reviews via supervised learning approaches (see Chapter 3 for more detail). Interestingly, our baseline experiments on this task, described in Section 3.2.1, show that humans may not always have the best intuition for choosing discriminating words. While we did experiment with a set of different features in our early work (see Pang et al. (2002), or Section 3.2), as we pointed out in Chapter 1, our main focus was not on feature engineering. Dave, Lawrence, and Pennock (2003) also took the supervised learning approach and experimented with multiple datasets of product reviews. They examined an extensive array of more sophisticated features based

on linguistic knowledge or heuristics, coupled with various feature-selection and smoothing techniques. Depending on the specific experimental setting, results are mixed compared to applying standard machine technique to simple unigram features. For instance, while trigrams and bigrams can improve performance under certain settings, interesting features based on dependency parsers failed to improve performance on their test sets. Subsequently various other types of features or feature selection schemes have been explored (Mullen and Collier, 2004; Gamon, 2004; Matsumoto, Takamura, and Okumura, 2005; Edoardo M. Airoldi, 2006).

One interesting problem arising from the domain-specific nature of this problem in the context of supervised learning is how to adapt the classifier to a new domain. Dave, Lawrence, and Pennock (2003) touch on this problem in their "follow-on" task where they apply the classifier trained on the pre-assembled dataset to product reviews found in search results that consist of different types of documents. But they did not investigate the effect of training-test mis-match in detail. Aue and Gamon (2005b) explored different approaches to customize a sentiment classification system to a new target domain in the absence of large amounts of labeled data. The different types of data they considered range from lengthy movie reviews to short, phrase-level user feedback to web surveys. Due to the significant differences in these domains (different subject matter as well as different styles and lengths of writing), simply applying the classifier learned on data from one domain can barely outperform the baseline for another domain. In fact, with 100 or 200 labeled items in the target domain, an EM algorithm that utilizes in-domain unlabeled data and ignores out-of-domain data altogether outperforms the method based exclusively on (both in- and out-of-domain) labeled data. Read (2005) did not explicitly address this domain transfer problem, but did find standard ma-

chine learning techniques to be both domain-dependent (with domains ranging from movie reviews to newswire articles), and temporally-dependent (based on datasets spanning different ranges of time periods but written at least one year apart). He also explored an alternative source of labeled data: emoticons used in Usenet newsgroup postings. The idea was to extract texts around smile or frown emoticons and map the emoticons into polarity labels.

In contrast to approaches based on supervised or semi-supervised learning, Turney's (2002) work on classification of reviews applied a specific unsupervised learning technique based on the mutual information between document phrases and the words "excellent" and "poor", where the mutual information is computed using statistics on data gathered by a search engine. He worked with reviews on different subjects, reporting accuracies ranging over 64% (movie reviews) to 84% (automobile reviews). Beineke et al. (2004) experimented with an extension of Turney's method that also utilized a small set of labeled data. Most unsupervised approaches to document-level polarity classification involve automatic labeling of words or phrases by their sentiment polarity (or "semantic orientation", as it is commonly referred to in the literature). In fact, some studies are entirely about building such lexicons. Hatzivassiloglou and McKeown (1997) presented an early approach based on linguistic heuristics. The main idea there is to use information extracted from conjunctions between adjectives in a large corpus — e.g., if we see two adjectives being linked by *but*, this suggests that they are of opposite orientations; conversely, being linked by *and* could be an evidence for the adjectives having the same orientation. The task is then casted into a clustering problem with constraints based on these heuristics. Most later studies in this direction start with a small set of seed words (in some cases, just two words with opposite

polarity), and use these seed words to (sometimes incrementally) tag other words. Corpus-based approaches examine co-occurrences with seed words based on large collections of text (Turney, 2002; Turney and Littman, 2003; Yu and Hatzivassiloglou, 2003; Aue and Gamon, 2005a), or search for the context-dependent labels by taking into account local constraints (Popescu and Etzioni, 2005). Alternatively, people have looked into exploiting knowledge encoded in WordNet such as relations (synonymy, antonymy and hyponymy) and glosses (Kim and Hovy, 2004; Hu and Liu, 2004; Kamps et al., 2004; Takamura, Inui, and Okumura, 2005; Andreevskaia and Bergler, 2006)

Subjectivity detection Work in polarity classification often assume the incoming documents to be opinionated. For many applications, we may need to decide whether a given document contains subjective information or not, or identify which portions of the document are subjective. An early work in this direction by Hatzivassiloglou and Wiebe (2000) examined the effects of adjective orientation and gradability on sentence subjectivity. The goal was to tell whether a given sentence is subjective or not judging from the adjectives appearing in that sentence. A number of projects address sentence-level or sub-sentence-level subjectivity detection in different domains (Wiebe, Wilson, and Bell, 2001; Wiebe and Wilson, 2002; Yu and Hatzivassiloglou, 2003; Riloff and Wiebe, 2003; Pang and Lee, 2004; Beineke et al., 2004; Kim and Hovy, 2005a; Wilson, Wiebe, and Hoffmann, 2005). Wiebe et al. (2004) present a comprehensive examination of using different clues and features for recognizing subjectivity in text. Subjectivity detection at the document level is closely related to some studies in *genre* classification (see Section 2.2 for more detail). For instance, Yu and Hatzivassiloglou (2003) achieve high accu-

racy (97%) with a Naive Bayes classifier on a particular corpus consisting of Wall Street Journal articles, where the task is to distinguish articles under *News and Business* (facts) from articles under *Editorial and Letter to the Editor* (opinions). Work in this direction is not limited to the binary distinction between subjective and objective labels. Wilson, Wiebe, and Hwa (2004) addressed the problem of determining clause-level opinion strength (e.g., "how mad are you"). Recent work also considered the relations between word sense disambiguation and subjectivity (Wiebe and Mihalcea, 2006).

Opinion mining and summarization An opinion-bearing document may contain off-topic passages that the readers may not be interested in. Hence, one possible approach to extract sentiment polarity on a given topic only is to first apply a sentence-level classifier to identify topical sentences (Hurst and Nigam, 2004). For reviews that cover several aspects of the subject matter, it may be desirable to summarize the opinions grouped by the specific aspects addressed, and there has been work that explored identifying features and opinions associated with these features from product reviews (Yi et al., 2003; Hu and Liu, 2004; Kobayashi et al., 2004; Popescu and Etzioni, 2005; Yi and Niblack, 2005).

Perspectives and viewpoints Some early work on non-factual-based text analysis dealt with perspectives and viewpoints (Wiebe and Rapaport, 1988; Wiebe, 1994). The MPQA (Multi Perspective Question Answering) project focuses on question-answering tasks that require the ability to analyze opinions in text, so that answers to questions such as "Was the most recent presidential election in Zimbabwe regarded as a fair election?" can be extracted from collections of documents. Low-level opinion annotations were developed and evaluated, which facil-

itated the study of a number of interesting problems such as identifying opinion holder and analyzing opinions at phrase level (Cardie et al., 2003; Stoyanov et al., 2004; Breck and Cardie, 2004; Wilson, Wiebe, and Hwa, 2004; Choi et al., 2005; Wiebe, Wilson, and Cardie, 2005; Kim and Hovy, 2005b). There has also been work that focused on specific pairs of perspectives such as identifying Israeli versus Palestinian viewpoints (Lin et al., 2006).

Affects and emotions People have considered various affect types, or the six "universal" emotions (Ekman, 1982): anger, disgust, fear, happiness, sadness, and surprise (Subasic and Huettner, 2001; Liu, Lieberman, and Selker, 2003; Alm, Roth, and Sproat, 2005), as well as computational approaches for humor recognition and generation (Mihalcea and Strapparava, 2006). Many interesting aspects of text like "happiness" or "mood" are also being explored in the context of informal text resources such as weblogs (Nicolov et al., 2006).

2.2 Other non-factual information in text

Another related area of research that addresses non-topic-based categorization is that of determining the *genre* of texts (Karlgren and Cutting, 1994; Kessler, Nunberg, and Schütze, 1997; Stamatatos, Fakotakis, and Kokkinakis, 2000; Finn, Kushmerick, and Smyth, 2002; Lee and Myaeng, 2002; Finn and Kushmerick, 2006). Since subjective genres, such as "editorial", are often one of the possible categories, such work can be closely related to subjectivity detection. There has also been research that concentrates on classifying documents according to their *source* or *source style*, with statistically-detected stylistic variation (Biber, 1988) serving as an important cue. Authorship identification is perhaps the most salient

example — Mosteller and Wallace's (1984) classic Bayesian study of the authorship of the Federalist Papers is one well-known instance. Argamon-Engelson et al. (1998b) consider the related problem of identifying not the particular author of a text, but its publisher (e.g. the *New York Times* vs. *The Daily News*); the work of Kessler, Nunberg, and Schütze (1997) on determining a document's "brow" (e.g., high-brow vs. "popular", or low-brow) has similar goals. Several recent workshops are dedicated to style analysis in text (Argamon, 2003; Argamon, Karlgren, and Shanahan, 2005; Argamon, Karlgren, and Uzuner, 2006). Finally, Tomokiyo and Jones (2001) studied an additional variant of the problem: distinguishing native from non-native English speaker transcripts.

Chapter 3

Polarity classification[1]

3.1 Introduction

As mentioned above, today, very large amounts of information are available in online documents, and a growing portion of such information comes in the form of people's experiences and opinions. It would be helpful for companies, recommender systems, and review or editorial sites to automatically compile digests of such information. It has proven quite useful in such contexts to create summaries of people's experiences and opinions that consist of subjective expressions extracted from reviews (as is commonly done in movie ads) or even just a review's *polarity* — positive ("thumbs up") or negative ("thumbs down").

This chapter describes our work on polarity classification, where we label an opinionated document as either positive or negative according to the author's overall opinion towards the subject matter expressed through the text. As we briefly discussed in Chapter 1, a challenging aspect of this problem that seems to dis-

[1]This chapter is based on the work described in "Thumbs up? Sentiment Classification using Machine Learning Techniques" with Lillian Lee and Shivakumar Vaithyanathan, which appeared in the proceedings of EMNLP 2002 (Pang, Lee, and Vaithyanathan, 2002) and "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" with Lillian Lee, which appeared in the proceedings of ACL 2004 (Pang and Lee, 2004).

tinguish it from traditional topic-based classification is that while topics are often identifiable by keywords alone, sentiment can be expressed more implicitly. For example, the sentence "how could anyone sit through that movie?" contains no single word that is clearly negative (see Section 3.2.4 for more examples). Thus, sentiment seems to require more *understanding* than the usual topic-based classification.

We first present our results obtained via machine learning techniques, treating this problem as a standard text classification problem; we also analyze the problem to gain a better understanding of how difficult it is. We then propose an approach where we first remove objective sentences from the input document via an automatic subjectivity classifier and apply a standard machine-learning classifier to the resulting *subjective extract*. This allows the classifier to focus on the portions of the document that actually express sentiment; at the same time, this provides a summary of the sentiment-oriented content of the document that can be potentially useful to the users as well. In particular, we experiment with an algorithm based on graph formulation that incorporates intra-document relationships.

3.1.1 Movie Review Data

The work described in this chapter is based on two corpora we developed (Pang, Lee, and Vaithyanathan, 2002; Pang and Lee, 2004). We have made these corpora (and other related datasets described in this thesis) publicly available[2], and these datasets have been adopted by more than a dozen other projects[3]. These corpora are in the movie review domain, which is experimentally convenient because there are large on-line collections of such reviews, often accompanied by information

[2]http://www.cs.cornell.edu/People/pabo/movie-review-data.html
[3]http://www.cs.cornell.edu/People/pabo/movie-review-data/otherexperiments.html

that provides machine-extractable labels. It is important to point out that our methods are in no sense specific to this domain, and should be easily applicable to other domains as long as sufficient training data exists. In addition, movie reviews were found to be more difficult to classify than other product reviews (Turney, (2002); but cf. Aue and Gamon (2005b), where movie reviews would found to be easier than book reviews and user feedback). We next describe these corpora in more detail. For other datasets suitable for work in sentiment analysis that we have collected and distributed, see Section 4.4 for a description on a scale dataset with a collection of documents whose labels come from a rating scale, and Section 5.2 for more details on a political speech dataset containing a set of congressional speeches with polarity labels (support or oppose).

Polarity Dataset The polarity dataset is a set of movie reviews with polarity labels which we created for our work on polarity classification. In many movie-review sites, reviewers summarize their overall sentiment with a machine-extractable *rating* indicator, such as a number of stars; hence, we did not need to hand-label the data for supervised learning or evaluation purposes.

Our data source was the Internet Movie Database (IMDb) archive of the `rec.arts.movies.reviews` newsgroup.[4] We selected only reviews where the author rating was expressed either with stars or some numerical value (other conventions varied too widely to allow for automatic processing). Ratings were automatically extracted and converted into one of three categories: positive, negative, or neutral. As the work described in this chapter was among the early attempts at automatically classifying documents based on sentiment, we did not have a firm grip of the difficulty of the task, and thus concentrated only on discriminating be-

[4] http://reviews.imdb.com/Reviews/

tween positive and negative sentiment as a reasonable first step. In fact, given the subtle nature of how sentiment is expressed in text, even two-class classification problem may appear quite daunting

See Chapter 4 for our experiments with all three categories as well as even more fine-grained classification.

To avoid domination of the corpus by a small number of prolific reviewers, we imposed a limit of fewer than 20 reviews per author per sentiment category, yielding a corpus of 752 negative and 1301 positive reviews, with a total of 144 reviewers represented. Version 1.0 of the polarity dataset (Pang, Lee, and Vaithyanathan, 2002) consists of 700 negative and 700 positive reviews randomly selected from that corpus. We later expanded this collection through an enhanced rating extraction method that covers a broader range of rating expressions and uses a more reliable conversion scheme. This resulted in Version 2.0 of the polarity dataset, which consists of 1000 positive and 1000 negative reviews (all written before 2002), with a total of 312 authors represented (Pang and Lee, 2004).

Subjectivity Dataset This dataset consists of sentences together with subjectivity labels ("subjective" vs "objective"). Riloff and Wiebe (2003) state that "It is [very hard] to obtain collections of individual sentences that can be easily identified as subjective or objective"; the polarity-dataset sentences, for example, have not been so annotated. Fortunately, we were able to mine the Web to create a large, automatically-labeled sentence corpus. To gather subjective sentences (or phrases), we collected 5000 movie-review *snippets* (e.g., "bold, imaginative, and impossible to resist") from www.rottentomatoes.com. To obtain (mostly) objective data, we took 5000 sentences from plot summaries available from the Internet

Movie Database (www.imdb.com) [5]. We only selected sentences or snippets at least ten words long and drawn from reviews or plot summaries of movies released post-2001, which prevents overlap with the polarity dataset.

3.2 Polarity Classification via Machine Learning Methods

This section presents our work on polarity classification with standard machine learning methods using version 1.0 of the polarity dataset.

3.2.1 A Closer Look At the Problem

At the time this work begun, intuitions seemed to differ as to the difficulty of the sentiment detection problem. An expert on using machine learning for text categorization predicted relatively low performance for automatic methods. On the other hand, it seems that distinguishing positive from negative reviews is relatively easy for humans, especially in comparison to the standard text categorization problem, where topics can be closely related. One might also suspect that there are certain words people tend to use to express strong opinions, so that it might suffice to simply produce a list of such words by introspection and rely on them alone to classify the texts. Indeed, this promise seems to underly early work on using manually-treated lexicons of indicative terms (Huettner and Subasic, 2000; Das and Chen, 2001).

To test this latter hypothesis, we asked two graduate students in computer science to (independently) choose good indicator words for positive and negative sentiments in movie reviews. Their selections, shown in Figure 3.1, seem intuitively

[5]These sentences (snippets) are randomly selected from the 5856 objective sentences (from plot summaries for movies released in 2002 and 2003) and 12763 snippets (from reviews for 2002 movies) we collected, where a total of 812 authors from 303 sources were represented.

	Proposed word lists	Accuracy	Ties
Human 1	positive: *dazzling, brilliant, phenomenal, excellent, fantastic* negative: *suck, terrible, awful, unwatchable, hideous*	58%	75%
Human 2	positive: *gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting* negative: *bad, cliched, sucks, boring, stupid, slow*	64%	39%

Figure 3.1: Baseline results for human word lists. Data: 700 positive and 700 negative reviews.

	Proposed word lists	Accuracy	Ties
Human 3	positive: *love, wonderful, best, great, superb, still, beautiful* negative: *bad, worst, stupid, waste, boring, ?, !*	69%	16%

Figure 3.2: Corresponding baselines using introspection and simple statistics of the *test* data.

plausible. We then converted their responses into simple decision procedures that essentially count the number of the proposed positive and negative words in a given document. We applied these procedures to a dataset in which the random-choice baseline result would be 50%.

As shown in Figure 3.1, the accuracy — percentage of documents classified correctly — for the human-based classifiers were 58% and 64%, respectively.[6] Note that the tie rates — percentage of documents where the two sentiments were rated equally likely — are quite high[7] (we chose a tie breaking policy that maximized the accuracy of these baselines).

[6]Later experiments using these words as features for machine learning methods did not yield better results.

[7]This is largely due to 0-0 ties.

While the tie rates suggest that the brevity of the human-produced lists is a factor in the relatively poor performance results, it is not the case that size alone necessarily limits accuracy. Based on a very preliminary examination of frequency counts in the entire corpus (including *test* data) plus introspection, we created a list of seven positive and seven negative words (including punctuation), shown in Figure 3.2. As that figure indicates, using these words raised the accuracy to 69%. Also, although this third list is of comparable length to the other two, it has a much lower tie rate of 16%. We further observe that some of the items in this third list, such as "?" or "still", would probably not have been proposed as possible candidates merely through introspection, although upon reflection one sees their merit (the question mark tends to occur in sentences like "What was the director thinking?"; "still" appears in sentences like "Still, though, it was worth seeing").

We conclude from these preliminary experiments that it is worthwhile to explore corpus-based techniques, rather than relying on prior intuitions, to select good indicator features and to perform sentiment classification in general. These experiments also provide us with baselines for experimental comparison; in particular, the third baseline of 69% might actually be considered somewhat difficult to beat, since it was achieved by examination of the test data (although our examination was rather cursory; we do not claim that there is no better-performing set of fourteen words).

3.2.2 Machine Learning Methods

Our aim in this work was to examine whether it suffices to treat sentiment classification simply as a special case of text categorization (with the two "topics" being positive sentiment and negative sentiment), or whether special sentiment-

categorization methods need to be developed. We experimented with three standard algorithms: Naive Bayes classification, maximum entropy classification, and support vector machines. The philosophies behind these three algorithms are quite different, but each has been shown to be effective in previous text categorization studies.

To implement these machine learning algorithms on our document data, we used the following standard bag-of-features framework. Let $\{f_1, \ldots, f_m\}$ be a predefined set of m *features* that can appear in a document; examples include the word "still" or the bigram "really stinks". Let $n_i(d)$ be the number of times f_i occurs in document d. Then, each document d is represented by the document vector $\vec{d} := (n_1(d), n_2(d), \ldots, n_m(d))$.

Naive Bayes

One approach to text classification is to assign to a given document d the class $C^* = \arg\max_C P(C \mid d)$. We derive the *Naive Bayes* (NB) classifier by first observing that by Bayes' rule,

$$P(C \mid d) = \frac{P(C)P(d \mid C)}{P(d)},$$

where $P(d)$ plays no role in selecting C^*. To estimate the term $P(d \mid C)$, Naive Bayes decomposes it by assuming the f_i's are conditionally independent given d's class:

$$P_{\text{NB}}(C \mid d) := \frac{P(C)\left(\prod_{i=1}^{m} P(f_i \mid C)^{n_i(d)}\right)}{P(d)}.$$

Our training method consists of relative-frequency estimation of $P(C)$ and $P(f_i \mid C)$, using add-one smoothing.

Despite its simplicity and the fact that its conditional independence assumption clearly does not hold in real-world situations, Naive Bayes-based text categoriza-

tion still tends to perform surprisingly well (Lewis, 1998); indeed, Domingos and Pazzani (1997) show that Naive Bayes is optimal for certain problem classes with highly dependent features. On the other hand, more sophisticated algorithms might (and often do) yield better results; we examine two such algorithms next.

Maximum Entropy

Maximum entropy classification (MaxEnt, or ME, for short) is an alternative technique which has proven effective in a number of natural language processing applications (Berger, Della Pietra, and Della Pietra, 1996). Nigam, Lafferty, and McCallum (1999) show that it sometimes, but not always, outperforms Naive Bayes at standard text classification. Its estimate of $P(C \mid d)$ takes the following exponential form:

$$P_{\text{ME}}(C \mid d) := \frac{1}{Z(d)} \exp\left(\sum_i \lambda_{i,C} F_{i,C}(d, C)\right) ,$$

where $Z(d)$ is a normalization function. $F_{i,C}$ is a *feature/class function* for feature f_i and class C, defined as follows:[8]

$$F_{i,C}(d, C') := \begin{cases} 1, & n_i(d) > 0 \;\; and \;\; C' = C \\ 0 & otherwise \end{cases}$$

For instance, a particular feature/class function might fire if and only if the bigram "still hate" appears and the document's sentiment is hypothesized to be negative.[9] Importantly, unlike Naive Bayes, MaxEnt makes no assumptions about the relationships between features, and so might potentially perform better when conditional independence assumptions are not met.

[8] We use a restricted definition of feature/class functions so that MaxEnt relies on the same sort of feature information as Naive Bayes.

[9] The dependence on class is necessary for parameter induction. See Nigam, Lafferty, and McCallum (1999) for additional motivation.

The $\lambda_{i,C}$'s are feature-weight parameters; inspection of the definition of P_{ME} shows that a large $\lambda_{i,C}$ means that f_i is considered a strong indicator for class C. The parameter values are set so as to maximize the entropy of the induced distribution (hence the classifier's name) subject to the constraint that the expected values of the feature/class functions with respect to the model are equal to their expected values with respect to the training data: the underlying philosophy is that we should choose the model making the fewest assumptions about the data while still remaining consistent with it, which makes intuitive sense. We use ten iterations of the improved iterative scaling algorithm (Della Pietra, Della Pietra, and Lafferty, 1997) for parameter training (this was a sufficient number of iterations for convergence of training-data accuracy), together with a Gaussian prior to prevent overfitting (Chen and Rosenfeld, 2000).

Support Vector Machines

Support vector machines (SVMs) have been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes (Joachims, 1998). They are *large-margin*, rather than probabilistic, classifiers, in contrast to Naive Bayes and MaxEnt. In the two-category case, the basic idea behind the training procedure is to find a hyperplane, represented by vector $\vec{w}$, that not only separates the document vectors in one class from those in the other, but for which the separation, or *margin*, is as large as possible. This search corresponds to a constrained optimization problem; letting $C_j \in \{1, -1\}$ (corresponding to positive and negative) be the correct class of document d_j, the solution can be written as

$$\vec{w} = \sum_j \alpha_j C_j \vec{d_j}, \quad \alpha_j \geq 0;$$

the α_j's are obtained by solving a dual optimization problem. Those $\vec{d_j}$ such that α_j is greater than zero are called *support vectors*, since they are the only document vectors contributing to $\vec{w}$. Classification of test instances consists simply of determining which side of $\vec{w}$'s hyperplane they fall on.

We used Joachim's (1999) SVM^{light} package[10] for training and testing, with all parameters set to their default values, after first length-normalizing the document vectors, as is standard (neglecting to normalize generally hurt performance slightly).

3.2.3 Evaluation

Experimental Set-up

We used documents from the polarity dataset (version 1.0) described in Section 3.1.1. We divided this data into three equal-sized folds, maintaining balanced class distributions in each fold. (We did not use a larger number of folds due to the slowness of the MaxEnt training procedure.) All results reported below, as well as the baseline results from Section 3.2.1, are the average three-fold cross-validation results on this data (of course, the baseline algorithms had no parameters to tune).

To prepare the documents, we automatically removed the rating indicators and extracted the textual information from the original HTML document format, treating punctuation marks as separate lexical items. No stemming or stoplists were used.

One unconventional step we took was to attempt to model the potentially important contextual effect of negation: clearly "good" and "not very good" indicate opposite sentiment orientations. Adapting a technique of Das and Chen (2001),

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

we added the tag NOT_ to every word between a negation word ("not", "isn't", "didn't", etc.) and the first punctuation mark following the negation word. (Preliminary experiments indicate that removing the negation tag had a negligible, but on average slightly harmful, effect on performance.)

For this study, we focused on features based on unigrams (with negation tagging) and bigrams. Because training MaxEnt is expensive in the number of features, we limited consideration to (1) the 16165 unigrams appearing at least four times in our 1400-document corpus (lower count cutoffs did not yield significantly different results), and (2) the 16165 bigrams occurring most often in the same data (the selected bigrams all occurred at least seven times). Note that we did not add negation tags to the bigrams, since we consider bigrams (and n-grams in general) to be an orthogonal way to incorporate context.

Results

Initial unigram results The classification accuracies resulting from using only unigrams as features are shown in line (1) of Table 3.1. As a whole, the machine learning algorithms clearly surpass the random-choice baseline of 50%. They also handily beat our two human-selected-unigram baselines of 58% and 64%, and, furthermore, perform well in comparison to the 69% baseline achieved via limited access to the test-data statistics, although the improvement in the case of SVMs is not so large.

On the other hand, in topic-based classification, all three classifiers have been reported to use bag-of-unigram features to achieve accuracies of 90% and above for particular categories (Joachims, 1998; Nigam, Lafferty, and McCallum, 1999)[11]

[11] Joachims (1998) used stemming and stoplists; in some of their experiments, Nigam, Lafferty, and McCallum (1999), like us, did not.

Table 3.1: Average three-fold cross-validation accuracies, in percent. Boldface: best performance for a given setting (row). Recall that our baseline results ranged from 50% to 69%.

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

— and such results are for settings with more than two classes. This provides suggestive evidence that sentiment categorization is more difficult than topic classification, which corresponds to the intuitions of the text categorization expert mentioned above.[12] Nonetheless, we still wanted to investigate ways to improve our sentiment categorization results; these experiments are reported below.

Feature frequency vs. presence Recall that we represent each document d by a feature-count vector $(n_1(d), \ldots, n_m(d))$. However, the definition of the MaxEnt feature/class functions $F_{i,c}$ only reflects the presence or absence of a feature, rather

[12]We could not perform the natural experiment of attempting topic-based categorization on our data because the only obvious topics would be the film being reviewed; unfortunately, in our data, the maximum number of reviews per movie is 27, too small for meaningful results.

than directly incorporating feature frequency. In order to investigate whether reliance on frequency information could account for the higher accuracies of Naive Bayes and SVMs, we binarized the document vectors, setting $n_i(d)$ to 1 if and only feature f_i appears in d, and reran Naive Bayes and SVM^{light} on these new vectors.[13]

As can be seen from line (2) of Table 3.1, better performance (*much* better performance for SVMs) is achieved by accounting only for feature presence, not feature frequency. Interestingly, this is in direct opposition to the observations of McCallum and Nigam (1998) with respect to Naive Bayes topic classification. We speculate that this indicates a difference between sentiment and topic categorization — perhaps due to topic being conveyed mostly by particular content words that tend to be repeated — but this remains to be verified. In any event, as a result of this finding, we did not incorporate frequency information into Naive Bayes and SVMs in any of the following experiments.

Bigrams In addition to looking specifically for negation words in the context of a word, we also studied the use of bigrams to capture more context in general. Note that bigrams and unigrams are surely not conditionally independent, meaning that the feature set they comprise violates Naive Bayes' conditional-independence assumptions; on the other hand, recall that this does not imply that Naive Bayes will necessarily do poorly (Domingos and Pazzani, 1997).

Line (3) of the results table shows that bigram information does not improve performance beyond that of unigram presence, although adding in the bigrams does

[13] Alternatively, we could have tried integrating frequency information into MaxEnt. However, feature/class functions are traditionally defined as binary (Berger, Della Pietra, and Della Pietra, 1996); hence, explicitly incorporating frequencies would require different functions for each count (or count bin), making training impractical. But cf. Nigam, Lafferty, and McCallum (1999).

not seriously impact the results, even for Naive Bayes. This would not rule out the possibility that bigram presence is as equally useful a feature as unigram presence; in fact, Pedersen (2001) found that bigrams alone can be effective features for word sense disambiguation. However, comparing line (4) to line (2) shows that relying just on bigrams causes accuracy to decline by as much as 5.8 percentage points. Hence, if context is in fact important, as our intuitions suggest, bigrams are not effective at capturing it in our setting.

Parts of speech We also experimented with appending POS tags to every word via Oliver Mason's Qtag program.[14] This serves as a crude form of word sense disambiguation (Wilks and Stevenson, 1998): for example, it would distinguish the different usages of "love" in "I love this movie" (indicating sentiment orientation) versus "This is a love story" (neutral with respect to sentiment). However, the effect of this information seems to be a wash: as depicted in line (5) of Table 3.1, the accuracy improves slightly for Naive Bayes but declines for SVMs, and the performance of MaxEnt is unchanged.

Since adjectives have been a focus of previous work in sentiment detection (Hatzivassiloglou and Wiebe, 2000; Turney, 2002), we looked at the performance of using adjectives alone.[15] Intuitively, we might expect that adjectives carry a great deal of information regarding a document's sentiment; indeed, the human-produced lists from Section 3.2.1 contain almost no other parts of speech. Yet, the results, shown in line (6) of Table 3.1, are relatively poor: the 2633 adjectives provide less useful information than unigram presence. Indeed, line (7) shows that simply using the 2633 most frequent unigrams is a better choice, yielding

[14] http://www.english.bham.ac.uk/staff/oliver/software/tagger/index.htm

[15] Turney's (2002) unsupervised algorithm uses bigrams containing an adjective or an adverb.

performance comparable to that of using (the presence of) all 16165 (line (2)). This may imply that applying explicit feature-selection algorithms on unigrams could improve performance.

Position An additional intuition we had was that the position of a word in the text might make a difference: movie reviews, in particular, might begin with an overall sentiment statement, proceed with a plot discussion, and conclude by summarizing the author's views. As a rough approximation to determining this kind of structure, we tagged each word according to whether it appeared in the first quarter, last quarter, or middle half of the document[16]. The results (line (8)) didn't differ greatly from using unigrams alone, but more refined notions of position might be more successful.

3.2.4 Discussion

The results produced via machine learning techniques are quite good in comparison to the human-generated baselines discussed in Section 3.2.1. In terms of relative performance, Naive Bayes tends to do the worst and SVMs tend to do the best, although the differences aren't very large.

On the other hand, we were not able to achieve accuracies on the sentiment classification problem comparable to those reported for standard topic-based categorization, despite the several different types of features we tried. Unigram presence information turned out to be the most effective; in fact, none of the alternative features we employed provided consistently better performance once unigram presence was incorporated. Interestingly, though, the superiority of presence infor-

[16]We tried a few other settings, e.g. first third vs. last third vs middle third, and found them to be less effective.

mation in comparison to frequency information in our setting contradicts previous observations made in topic-classification work (McCallum and Nigam, 1998).

What accounts for these two differences — difficulty and types of information proving useful — between topic and sentiment classification, and how might we improve the latter? To answer these questions, we examined the data further. (All examples below are drawn from the full 2053-document corpus.)

Consider the word "good", which would at first appear to be a strong indicator for positive sentiment. However, it appears 478 times in positive reviews and 411 in negative reviews. The reason "good" occurs frequently in the negative class is not just because people use it in sentiment-neutral phrases. In many negative reviews, it is still being used in its subjective sense, as in the following "thwarted expectations" narrative, where the author sets up a deliberate contrast to earlier discussion: for example,

> *This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up.*

We also have many instances of the converse occurrence:

> *Some horror films just get off on non-stop gore, cheap thrills, and, well, bad acting. "Alien" has none of these.*

In the latter example, the issue appears to be one of reference resolution: it would certainly help if we could recognize that "bad acting" is not referring to the subject matter of this review. Co-reference resolution is a hard enough problem in itself, but we also have example like this: or

> *I hate the Spice Girls. ...[3 things the author hates about them]... Why I saw this movie is a really, really, really long story, but I did, and one would think I'd despise every minute of it. But... Okay, I'm really ashamed of it, but I enjoyed it. I mean, I admit it's a really awful movie, ...the ninth floor of hell...The plot is such a mess that it's terrible. But I loved it.*

Note that this phenomenon is related to another common theme, that of "a good actor trapped in a bad movie":

> *AN AMERICAN WEREWOLF IN PARIS is a failed attempt... Julie Delpy is far too good for this movie. She imbues Serafine with spirit, spunk, and humanity. This isn't necessarily a good thing, since it prevents us from relaxing and enjoying AN AMERICAN WEREWOLF IN PARIS as a completely mindless, campy entertainment experience. Delpy's injection of class into an otherwise classless production raises the specter of what this film could have been with a better script and a better cast ... She was radiant, charismatic, and effective ....*

This example is particularly striking in that a human wouldn't need to read past the first sentence, but unfortunately, the techniques we used in our experiments will, and will potentially be fooled by the many accolades for the actor rather than the film itself.

In these examples, a human would easily detect the true sentiment of the review, but bag-of-features classifiers would presumably find these instances difficult, since there are many words indicative of the opposite sentiment to that of the entire review. Fundamentally, it seems that some form of discourse analysis is necessary

(using more sophisticated techniques than our positional feature mentioned above), or at least some way of determining the focus of each sentence, so that one can decide when the author is talking about the film itself. (Turney (2002) makes a similar point, noting that for reviews, "the whole is not necessarily the sum of the parts".) Furthermore, it seems likely that this thwarted-expectations rhetorical device will appear in many types of texts (e.g., editorials) devoted to expressing an overall opinion about some topic. Hence, we believe that an important next step is the identification of features indicating whether sentences are on-topic (which is a kind of co-reference problem); we look forward to addressing this challenge in future work.

3.3 Polarity classification with subjective summaries

In contrast to previous approaches that focused on selecting indicative lexical features (e.g., the word "good"), classifying a document according to the number of such features that occur anywhere within it, we proposed a two-step process that uses subjective summarizations to help polarity classification (1) label the sentences in the document as either subjective or objective, discarding the latter; and then (2) apply a standard machine-learning classifier to the resulting *extract*. This can prevent the polarity classifier from considering irrelevant or even potentially misleading text: for example, although the sentence "The protagonist tries to protect her good name" contains the word "good", it could well be embedded in a negative movie review. Also, as mentioned above, the extract itself can be provided to users as a summary of the subjective content of the document. In particular, we experimented with using a framework based on finding minimum cuts to model context information.

3.3.1 Architecture

One can consider document-level polarity classification to be just a special (more difficult) case of text categorization with sentiment- rather than topic-based categories. Hence, standard machine-learning classification techniques, such as support vector machines, can be applied to the entire documents themselves, as was done in Section 3.2. We refer to such classification techniques as *default polarity classifiers*.

However, as noted above, we may be able to improve polarity classification by removing objective sentences (such as plot summaries in a movie review). We therefore propose, as depicted in Figure 3.3, to first employ a *subjectivity detector* that determines whether each sentence is subjective or not: discarding the objective ones creates an *extract* that should better represent a review's subjective content to a default polarity classifier.

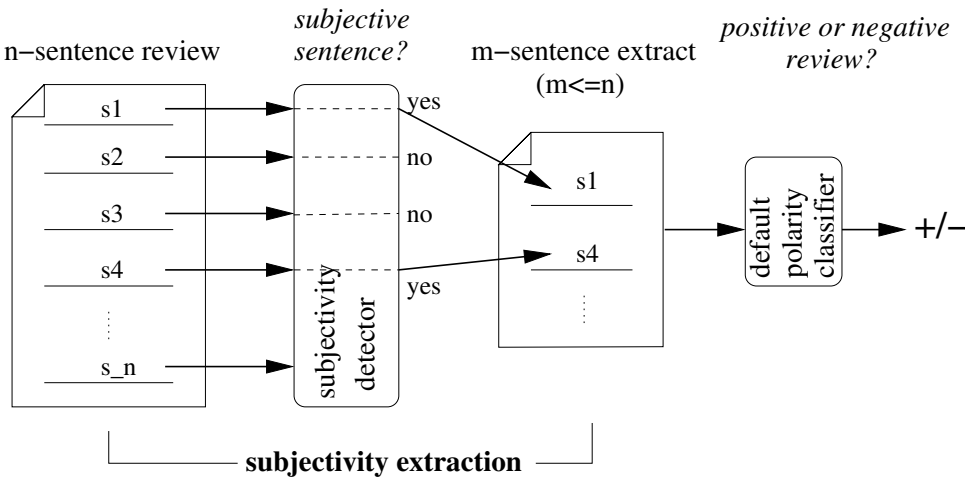


Figure 3.3: Polarity classification via subjectivity detection.

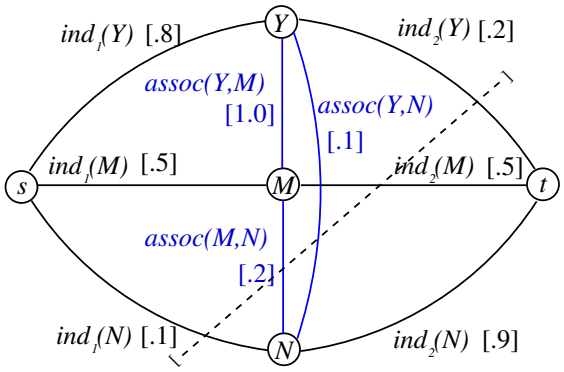
To our knowledge, previous work has not integrated sentence-level subjectivity detection with document-level sentiment polarity. Yu and Hatzivassiloglou (2003) provide methods for sentence-level analysis and for determining whether a docu-

ment is subjective or not, but do not combine these two types of algorithms or consider document polarity classification. The motivation behind the single-sentence selection method of Beineke et al. (2004) is to reveal a document's sentiment polarity, but they do not evaluate the polarity-classification accuracy that results.

3.3.2 Context and Subjectivity Detection

As with document-level polarity classification, we could perform subjectivity detection on individual sentences by applying a standard classification algorithm on each sentence in isolation. However, modeling proximity relationships between sentences would enable us to leverage *coherence*: text spans occurring near each other (within discourse boundaries) may share the same subjectivity status, other things being equal (Wiebe, 1994).

We would therefore like to supply our algorithms with pair-wise interaction information, e.g., to specify that two particular sentences should ideally receive the same subjectivity label — without stating which label this should be. This is difficult to do with classifiers whose input consists simply of *individual* feature vectors, such as naive Bayes. For example, one could try adding extra dummy features for which the two sentences have the same value, but what values for these features should all the other sentences have? Fortunately, we can overcome this problem using a graph-based formulation relying on finding *minimum cuts*. This was first pointed in the machine-learning literature by Blum and Chawla (2001), although they focused on similarity between items (the motivation being to combine labeled and unlabeled data), whereas we are concerned with physical proximity between the items to be classified.



C_1	Individual penalties	Association penalties	Cost
{Y,M}	$.2 + .5 + .1$	$.1 + .2$	1.1
(none)	$.8 + .5 + .1$	0	1.4
{Y,M,N}	$.2 + .5 + .9$	0	1.6
{Y}	$.2 + .5 + .1$	$1.0 + .1$	1.9
{N}	$.8 + .5 + .9$	$.1 + .2$	2.5
{M}	$.8 + .5 + .1$	$1.0 + .2$	2.6
{Y,N}	$.2 + .5 + .9$	$1.0 + .2$	2.8
{M,N}	$.8 + .5 + .9$	$1.0 + .1$	3.3

Figure 3.4: Graph for classifying three items. Brackets enclose example values; here, the individual scores happen to be probabilities. Based on *individual* scores alone, we would put Y ("yes") in C_1, N ("no") in C_2, and be undecided about M ("maybe"). But the *association* scores favor cuts that put Y and M in the same class, as shown in the table. Thus, the minimum cut, indicated by the dashed line, places M together with Y in C_1.

3.3.3 Cut-based classification

Figure 3.4 shows a worked example of the concepts in this section.

Suppose we have n items $x_1, \ldots, x_n$ to divide into two classes C_1 and C_2, and we have access to two types of information:

- *Individual* scores $ind_j(x_i)$: non-negative estimates of each x_i's preference for being in C_j based on just the features of x_i alone; and
- *Association* scores $assoc(x_i, x_k)$: non-negative estimates of how important it is that x_i and x_k be in the same class.[17]

[17] Asymmetry is allowed, but we used symmetric scores.

We would like to maximize each item's "net happiness": its individual score for the class it is assigned to, minus its individual score for the other class. But, we also want to penalize putting tightly-associated items into different classes. Thus, after some algebra, we arrive at the following optimization problem: assign the x_is to C_1 and C_2 so as to minimize the *partition cost*

$$\sum_{x \in C_1} ind_2(x) + \sum_{x \in C_2} ind_1(x) + \sum_{\substack{x_i \in C_1, \\ x_k \in C_2}} assoc(x_i, x_k).$$

The problem appears intractable, since there are 2^n possible binary partitions of the x_i's. However, suppose we represent the situation in the following manner. Build an undirected graph G with vertices $\{v_1, \ldots, v_n, s, t\}$; the last two are, respectively, the *source* and *sink*. Add n edges (s, v_i), each with weight $ind_1(x_i)$, and n edges (v_i, t), each with weight $ind_2(x_i)$. Finally, add $\binom{n}{2}$ edges (v_i, v_k), each with weight $assoc(x_i, x_k)$. Then, cuts in G are defined as follows:

Definition 1 *A* cut (S, T) *of* G *is a partition of its nodes into sets* $S = \{s\} \cup S'$ *and* $T = \{t\} \cup T'$*, where* $s \notin S', t \notin T'$*. Its* cost $cost(S, T)$ *is the sum of the weights of all edges crossing from* S *to* T*. A* minimum cut *of* G *is one of minimum cost.*

Observe that every cut corresponds to a partition of the items and has cost equal to the partition cost. Thus, our optimization problem reduces to finding minimum cuts.

Practical advantages As we have noted, formulating our subjectivity-detection problem in terms of graphs allows us to model item-specific and pairwise information independently. Note that this is a very flexible paradigm. For instance, it is perfectly legitimate to use knowledge-rich algorithms employing deep linguistic knowledge about sentiment indicators to derive the individual scores. And we could

also simultaneously use knowledge-lean methods to assign the association scores. Interestingly, Yu and Hatzivassiloglou (2003) compared an individual-preference classifier against a relationship-based method, but didn't combine the two; the ability to *coordinate* such algorithms is precisely one of the strengths of our approach.

But a crucial advantage specific to the utilization of a minimum-cut-based approach is that we can use *maximum-flow* algorithms with polynomial asymptotic running times — and near-linear running times in practice — to *exactly* compute the minimum-cost cut(s), despite the apparent intractability of the optimization problem (Cormen, Leiserson, and Rivest, 1990; Ahuja, Magnanti, and Orlin, 1993).[18] In contrast, other graph-partitioning problems that have been previously used to formulate NLP classification problems[19] are NP-complete (Hatzivassiloglou and McKeown, 1997; Agrawal et al., 2003; Joachims, 2003). In the next secion, We examine in the next more detail related work using graph-based methods.

3.3.4 Related work for the graph-cut formulation

As we mentioned earlier, our approach is inspired by Blum and Chawla (2001), who used minimum cuts as a means to integrate labeled and unlabeled data, reporting improvements over ID3 and 3-nearest-neighbors. Although they focused on similarity between items (the motivation being to combine labeled and unlabeled data), whereas we are concerned with physical proximity between the items to be classified; indeed, in computer vision, modeling proximity information via graph cuts has led to very effective classification (Boykov, Veksler, and Zabih, 1999), which is another source of inspiration for our work. In addition to our

[18] Code available at http://www.avglab.com/andrew/soft.html.

[19] Graph-based approaches to general *clustering* problems are too numerous to mention here.

own work described in this Chapter and Chapter 5, this framework has also been exploited for other NLP tasks (Agarwal and Bhattacharyya, 2005; Barzilay and Lapata, 2005; Malioutov and Barzilay, 2006). To our knowledge, the only prior uses of maximum-flow techniques in NLP are the work of Gaussier (1998) and Popescu, Etzioni, and Kautz (2003). But utilizing flow to compute *matchings* between items (e.g., French and English sentences in the first case), as these papers do, does not make use of the crucial information-integration advantage conferred by the minimum-cut formulation.

Work based on *normalized cuts* (Dhillon, 2001; Joachims, 2003) adds an extra "balanced-size" constraint to the mix; but finding the optimal solution in this new situation is NP-complete. This intractability also holds for *maximum cut*, used by Agrawal et al. (2003) to find "camps" in newsgroups. In addition to the work of Blum and Chawla (2001) and Joachims (2003), other notable early papers on graph-based semi-supervised learning include Bansal, Blum, and Chawla (2002) and Kondor and Lafferty (2002). Zhu (2005a) maintains a survey of this area.

Recently, several alternative, often quite sophisticated approaches to *collective classification* have been proposed (Neville and Jensen, 2000; Lafferty, McCallum, and Pereira, 2001; Getoor et al., 2002; Taskar, Abbeel, and Koller, 2002; Taskar, Guestrin, and Koller, 2003; Taskar, Chatalbashev, and Koller, 2004; McCallum and Wellner, 2004). It would be interesting to investigate the application of such methods to our problem. However, we also believe that our approach has important advantages, including conceptual simplicity and the fact that it is based on an underlying optimization problem that is provably and in practice easy to solve.

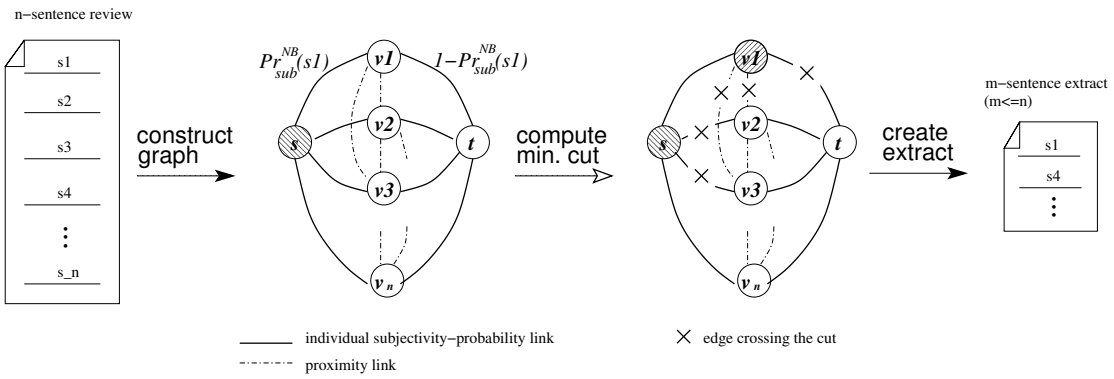


Figure 3.5: Graph-cut-based creation of subjective extracts.

3.3.5 Evaluation Framework

Data Our experiments involve classifying movie reviews as either positive or negative; for this task, we used the polarity dataset (version 2.0) In addition, to train our sentence-level subjectivity detectors, we used the subjectivity dataset (see Section 3.1.1 for more detail).

Default polarity classifiers We tested support vector machines (SVMs) and Naive Bayes (NB). Following the work described in Section 3.2, we use unigram-presence features: the ith coordinate of a feature vector is 1 if the corresponding unigram occurs in the input text, 0 otherwise. (For SVMs, the feature vectors are length-normalized). Each default document-level polarity classifier is trained and tested on the extracts formed by applying one of the sentence-level subjectivity detectors to reviews in the polarity dataset.

Subjectivity detectors As noted above, we can use our default polarity classifiers as "basic" sentence-level subjectivity detectors (after retraining on the subjectivity dataset) to produce extracts of the original reviews. We also create a family of cut-based subjectivity detectors; these take as input the *set* of sentences appear-

ing in a single document and determine the subjectivity status of all the sentences simultaneously using per-item and pairwise relationship information. Specifically, for a given document, we use the construction in Section 3.3.2 to build a graph wherein the source s and sink t correspond to the class of subjective and objective sentences, respectively, and each internal node v_i corresponds to the document's i^{th} sentence s_i. We can set the individual scores $ind_1(s_i)$ to $Pr^{NB}_{sub}(s_i)$ and $ind_2(s_i)$ to $1 - Pr^{NB}_{sub}(s_i)$, as shown in Figure 3.5, where $Pr^{NB}_{sub}(s)$ denotes Naive Bayes' estimate of the probability that sentence s is subjective; or, we can use the weights produced by the SVM classifier instead.[20] If we set all the association scores to zero, then the minimum-cut classification of the sentences is the same as that of the basic subjectivity detector. Alternatively, we can incorporate the degree of *proximity* between pairs of sentences, controlled by three parameters. The threshold T specifies the maximum distance two sentences can be separated by and still be considered proximal. The non-increasing function $f(d)$ specifies how the influence of proximal sentences decays with respect to distance d; in our experiments, we tried $f(d) = 1$, e^{1-d}, and $1/d^2$. The constant c controls the relative influence of the association scores: a larger c makes the minimum-cut algorithm more loath to put

[20] We converted any given SVM output d_i, which is a signed distance (negative=objective) from the separating hyperplane, to a non-negative number by

$$ind_1(s_i) \stackrel{def}{=} \begin{cases} 1 & d_i > 2; \\ (2 + d_i)/4 & -2 \leq d_i \leq 2; \\ 0 & d_i < -2. \end{cases}$$

and $ind_2(s_i) = 1 - ind_1(s_i)$. Note that scaling is employed only for consistency; the algorithm itself does not require probabilities for individual scores.

proximal sentences in different classes. With these in hand[21], we set (for $j > i$)

$$assoc(s_i, s_j) \stackrel{def}{=} \begin{cases} f(j-i) \cdot c & if (j-i) \leq T; \\ 0 & otherwise. \end{cases}$$

3.3.6 Experimental Results

Below, we report average accuracies computed by ten-fold cross-validation over the polarity dataset. Section 3.3.6 examines our basic subjectivity extraction algorithms, which are based on individual-sentence predictions alone. Section 3.3.6 evaluates the more sophisticated form of subjectivity extraction that incorporates context information via the minimum-cut paradigm.

As we will see, the use of subjectivity extracts can in the best case provide satisfying improvement in polarity classification, and otherwise can at least yield polarity-classification accuracies indistinguishable from employing the full review. At the same time, the extracts we create are both smaller on average than the original document and more effective as input to a default polarity classifier than the same-length counterparts produced by standard summarization tactics (e.g., first- or last-N sentences). We therefore conclude that subjectivity extraction produces effective summaries of document sentiment.

Basic subjectivity extraction

As noted in Section 3.3.5, both Naive Bayes and SVMs can be trained on our subjectivity dataset and then used as a basic subjectivity detector. The former has somewhat better average ten-fold cross-validation performance on the subjectivity

[21] Parameter training is driven by optimizing the performance of the downstream polarity classifier rather than the detector itself because the subjectivity dataset's sentences come from different reviews, and so are never proximal.

dataset (92% vs. 90%), and so for clarity, our initial discussions will focus on the results attained via NB subjectivity detection.

Employing Naive Bayes as a subjectivity detector ($Extract_{NB}$) in conjunction with a Naive Bayes document-level polarity classifier achieves 86.4% accuracy.[22] This is a clear improvement over the 82.8% that results when no extraction is applied (*Full review*); indeed, the difference is highly statistically significant ($p < 0.01$, paired t-test). With SVMs as the polarity classifier instead, the *Full review* performance rises to 87.15%, but comparison via the paired t-test reveals that this is statistically indistinguishable from the 86.4% that is achieved by running the SVM polarity classifier on $Extract_{NB}$ input. (More improvements to extraction performance are reported later in this section.)

These findings indicate[23] that the extracts preserve (and, in the NB polarity-classifier case, apparently clarify) the sentiment information in the originating documents, and thus are good summaries from the polarity-classification point of view. Further support comes from a "flipping" experiment: if we give as input to the default polarity classifier an extract consisting of the sentences labeled *objective*, accuracy drops dramatically to 71% for NB and 67% for SVMs. This confirms our hypothesis that sentences discarded by the subjectivity extraction process are indeed much less indicative of sentiment polarity.

Moreover, the subjectivity extracts are much more compact than the original documents (an important feature for a summary to have): they contain on average only about 60% of the source reviews' words. (This *word preservation rate* is plotted along the x-axis in the graphs in Figure 3.7.) This prompts us to study

[22] This result and others are depicted in Figure 3.7; for now, consider only the y-axis in those plots.

[23] Recall that direct evidence is not available because the polarity dataset's sentences lack subjectivity labels.

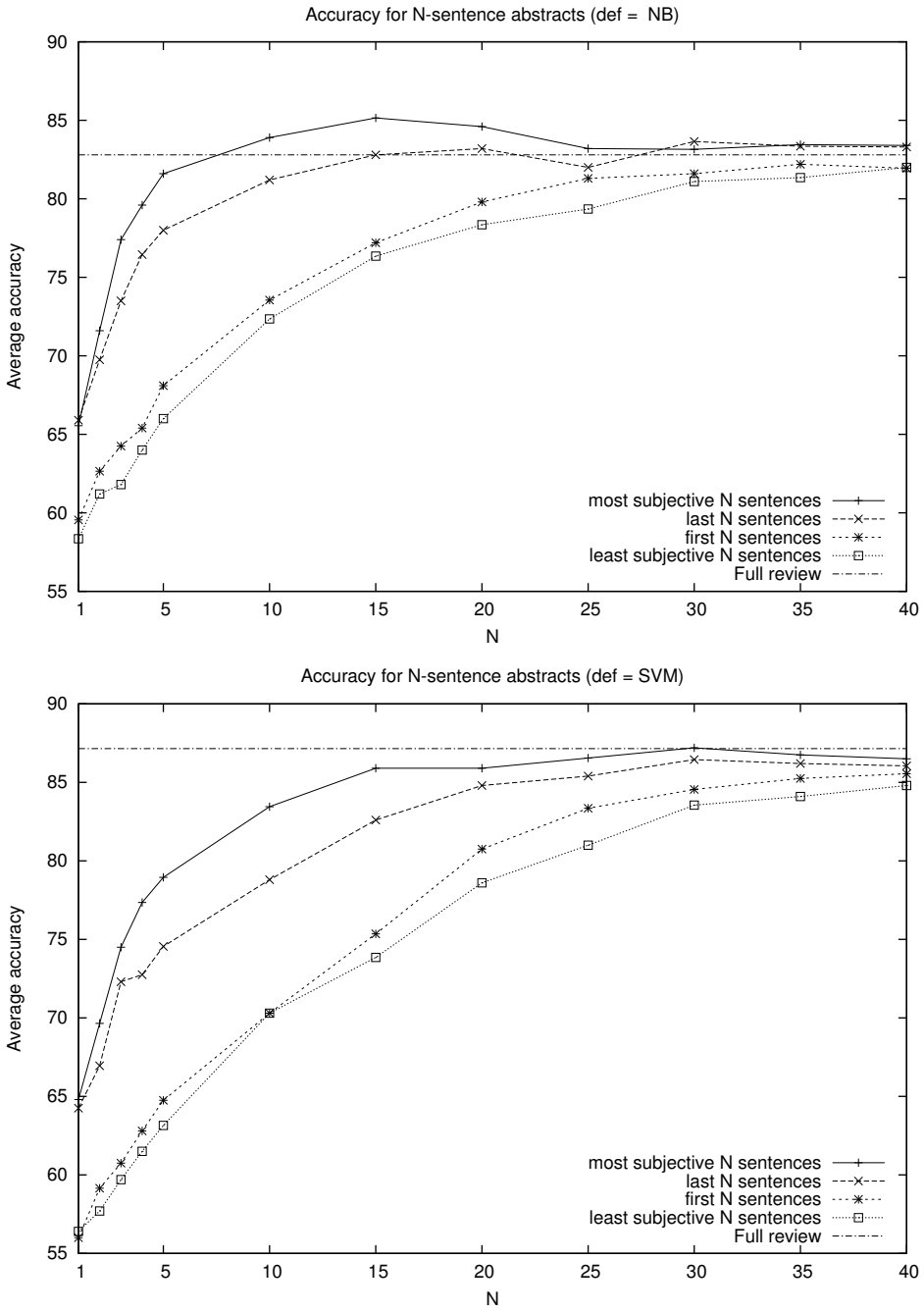


Figure 3.6: Accuracies using N-sentence extracts for NB (up) and SVM (down) default polarity classifiers.

how much reduction of the original documents subjectivity detectors can perform and still accurately represent the texts' sentiment information.

We can create subjectivity extracts of varying lengths by taking just the N *most subjective* sentences[24] from the originating review. As one baseline to compare against, we take the canonical summarization standard of extracting the *first* N sentences — in general settings, authors often begin documents with an overview. We also consider the *last* N sentences: in many documents, concluding material may be a good summary, and www.rottentomatoes.com tends to select "snippets" from the end of movie reviews (Beineke et al., 2004). Finally, as a sanity check, we include results from the N *least subjective* sentences according to Naive Bayes.

Figure 3.6 shows the polarity classifier results as N ranges between 1 and 40. Our first observation is that the NB detector provides very good "bang for the buck": with subjectivity extracts containing as few as 15 sentences, accuracy is quite close to what one gets if the entire review is used. In fact, for the NB polarity classifier, just using the 5 most subjective sentences is almost as informative as the *Full review* while containing on average only about 22% of the source reviews' words.

Also, it so happens that at $N = 30$, performance is actually slightly better than (but statistically indistinguishable from) *Full review* even when the SVM default polarity classifier is used (87.2% vs. 87.15%).[25] This suggests potentially effective extraction alternatives other than using a fixed probability threshold (which resulted in the lower accuracy of 86.4% reported above).

Furthermore, we see in Figure 3.6 that the N most-subjective-sentences method

[24]These are the N sentences assigned the highest probability by the basic NB detector, regardless of whether their probabilities exceed 50% and so would actually be classified as subjective by Naive Bayes. For reviews with less than N sentences, the entire review is selected.

[25]Note that roughly half of the documents in the polarity dataset contain more than 30 sentences (average=32.3, standard deviation 15).

generally outperforms the other baseline summarization methods (which perhaps suggests that sentiment summarization cannot be treated the same as topic-based summarization, although this conjecture would need to be verified on other domains and data). It's also interesting to observe how much better the last N sentences are than the first N sentences; this may reflect a (hardly surprising) tendency for movie-review authors to place plot descriptions at the beginning rather than the end of the text and conclude with overtly opinionated statements.

Incorporating context information

The previous section demonstrated the value of subjectivity detection. We now examine whether context information, particularly regarding sentence proximity, can further improve subjectivity extraction. As discussed in Section 3.3.2 and 3.3.5, contextual constraints are easily incorporated via the minimum-cut formalism but are difficult to encode into the input for standard Naive Bayes and SVMs.

Figure 3.7 shows the effect of adding in proximity information. $Extract_{NB+Prox}$ and $Extract_{SVM+Prox}$ are the graph-based subjectivity detectors using Naive Bayes and SVMs, respectively, for the individual scores; we depict the best performance achieved by a single setting of the three proximity-related edge-weight parameters over all ten data folds[26] (parameter selection was not a focus of the current work). The two comparisons we are most interested in are $Extract_{NB+Prox}$ versus $Extract_{NB}$ and $Extract_{SVM+Prox}$ versus $Extract_{SVM}$.

We see that the context-aware graph-based subjectivity detectors tend to create extracts that are more informative (statistically significant so (paired t-test) for SVM subjectivity detectors only), although these extracts are longer than their

[26] Parameters are chosen from $T \in \{1, 2, 3\}$, $f(d) \in \{1, e^{1-d}, 1/d^2\}$, and $c \in [0, 1]$ at intervals of 0.1.

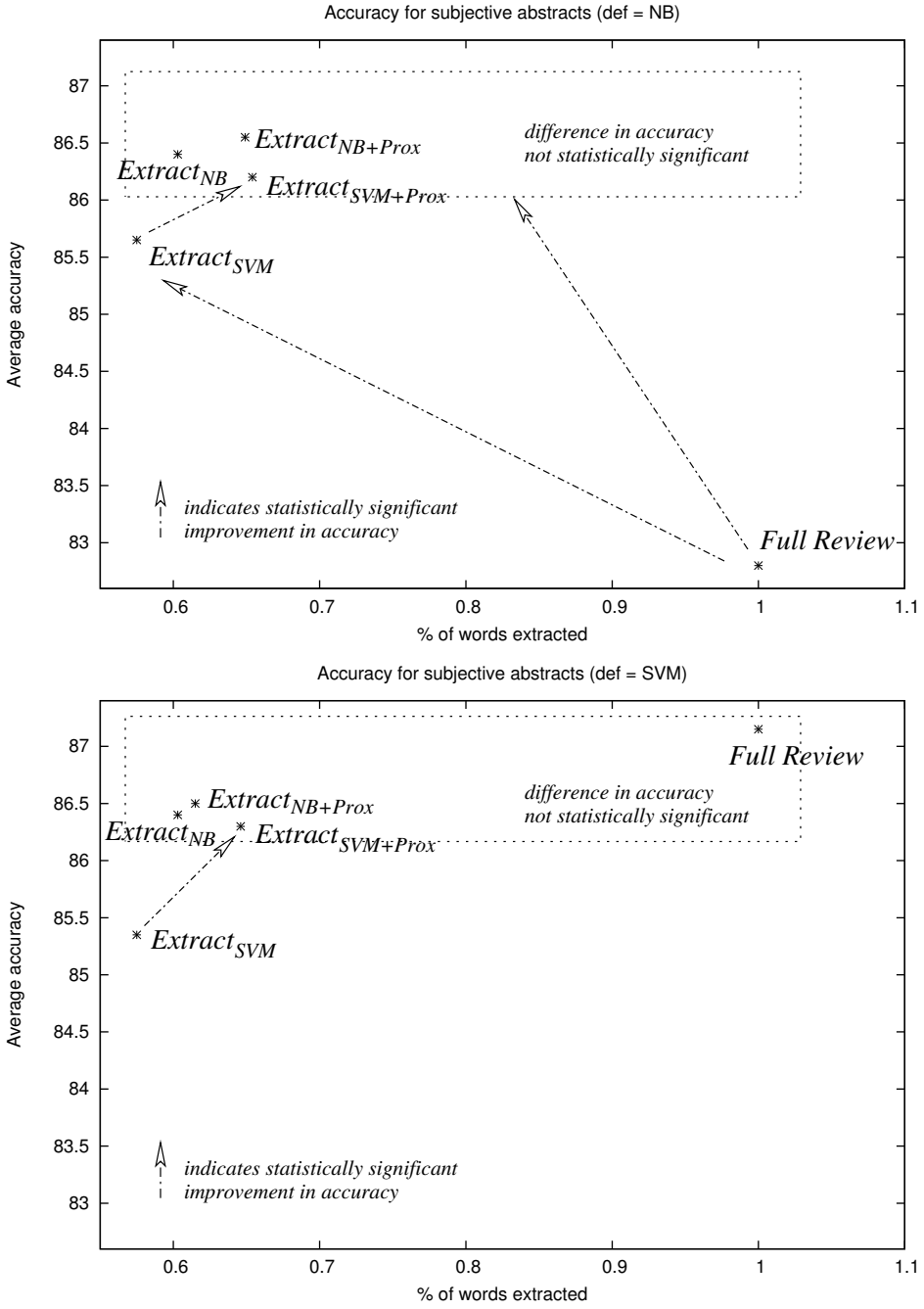


Figure 3.7: Word preservation rate vs. accuracy, NB (up) and SVMs (down) as default polarity classifiers. Also indicated are results for some statistical significance tests.

context-blind counterparts. We note that the performance enhancements cannot be attributed entirely to the mere inclusion of more sentences regardless of whether they are subjective or not — one counterargument is that *Full review* yielded substantially worse results for the NB default polarity classifier— and at any rate, the graph-derived extracts are still substantially more concise than the full texts.

Now, while a bias for assigning nearby sentences to the same category seems quite difficult to incorporate into NB and SVM subjectivity detectors, we also wish to investigate whether our graph-based paradigm makes better use of contextual constraints that *can* be encoded into the input of standard classifiers. For illustrative purposes, we consider paragraph-boundary information (looking only at SVM subjectivity detection for simplicity's sake).

It seems intuitively plausible that paragraph boundaries (an approximation to discourse boundaries) loosen coherence constraints between nearby sentences. To capture this notion for minimum-cut-based classification, we can simply reduce the association scores for all pairs of sentences that occur in different paragraphs by multiplying them by a cross-paragraph-boundary weight $w \in [0, 1]$. For standard classifiers, we can employ the trick of having the detector treat paragraphs, rather than sentences, as the basic unit to be labeled. This enables the standard classifier to utilize coherence between sentences in the same paragraph; on the other hand, it also (probably unavoidably) poses a hard constraint that all of a paragraph's sentences get the same label, which increases noise sensitivity.[27] Our experiments reveal the graph-cut formulation as the better approach: for both default polarity classifiers (NB and SVM), some choice of parameters (including w) for $Extract_{SVM+Prox}$ yields statistically significant improvement over

[27]For example, in the data we used, boundaries may have been missed due to malformed html.

its paragraph-unit non-graph counterpart (NB: 86.4% vs. 85.2%; SVM: 86.15% vs. 85.45%).

3.3.7 Classification based on hidden Markov models

Labeling each sentence in a document as either subjective or objective can also be treated as a sequence-labeling problem and modeled by hidden Markov models (Rabiner, 1989).

Recall our task is to divide n items $x_1, \ldots, x_n$ into two classes C_1 and C_2. These two classes can be represented as two states Q_1 and Q_2 in a hidden Markov model (HMM). The emission probabilities (i.e., the probability with which each state generates a given sentence in the sequence) can be obtained from language models trained on the subjectivity datasets. The transition probabilities can be determined from (unlabeled) data by optimizing the joint probability that the sequences are generated. Once these parameters are determined, the sentences can be labeled via the Viterbi algorithm. How does an HMM compare to the graph-cut formulation we just investigated Which model is more expressive for this task? Note that a first-order HMM can only capture relationships between adjacenct pairs (as opposed to arbitrary pairs of nearby sentences), and is thus similar to the graph-cut model with linear proximity function setting $T = 1$. In this sense, the HMM may be seen as a special case of the graph-cut formulation. On the other hand, our graph-cut formulation do not differentiate between transiting from the subjective state to the objective state and the other way around. We pay the same panelty for putting adjacent pairs into two classes, regardless of which label the first item in the pair receives. If it turns out that one state is more "sticky" than the other, then an HMM has the advantage in that it can easily model this difference

with different transition probabilities. However, in our preliminary experiments using naive Bayes models to provide the emission probabilities, we did not get improvement with HMMs over the graph-cut formulation.

3.3.8 Conclusions

We examined the relation between subjectivity detection and polarity classification, showing that subjectivity detection can compress reviews into much shorter extracts that still retain polarity information at a level comparable to that of the full review. In fact, for the Naive Bayes polarity classifier, the subjectivity extracts are shown to be more effective input than the originating document, which suggests that they are not only shorter, but also "cleaner" representations of the intended polarity.

We have also shown that employing the minimum-cut framework results in the development of efficient algorithms for sentiment analysis. Utilizing contextual information via this framework can lead to statistically significant improvement in polarity-classification accuracy. Directions for future research include developing parameter-selection techniques, incorporating other sources of contextual cues besides sentence proximity, and investigating other means for modeling such information.

Chapter 4

Exploiting class relationships for sentiment categorization with respect to rating scales[1]

4.1 Introduction

Most prior work on the specific problem of categorizing expressly opinionated text (including those presented in Chapter 3) has focused on the binary distinction of positive vs. negative (Turney, 2002; Pang, Lee, and Vaithyanathan, 2002; Dave, Lawrence, and Pennock, 2003; Yu and Hatzivassiloglou, 2003). But it is often helpful to have more information than this binary distinction provides, especially if one is ranking items by recommendation or comparing several reviewers' opinions: example applications include collaborative filtering and deciding which conference submissions to accept.

Therefore, in this chapter we consider generalizing to finer-grained *scales*: rather

[1]This chapter is based on the paper "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales" with Lillian Lee, which appeared in the proceedings of ACL 2005 (Pang and Lee, 2005).

than just determine whether a review is "thumbs up" or not, we attempt to infer the author's implied numerical rating, such as "three stars" or "four stars". Note that this differs from identifying opinion *strength* (Wilson, Wiebe, and Hwa, 2004): rants and raves have the same strength but represent opposite evaluations, and referee forms often allow one to indicate that one is very confident (high strength) that a conference submission is mediocre (middling rating). Also, our task differs from *ranking* not only because one can be given a single item to classify (as opposed to a set of items to be ordered relative to one another), but because there are settings in which classification is harder than ranking, and vice versa.

One can apply standard n-ary classifiers or regression to this *rating-inference problem*; independent work by Koppel and Schler (2005) considers such methods. But an alternative approach that explicitly incorporates information about item similarities together with label similarity information (for instance, "one star" is closer to "two stars" than to "four stars") is to think of the task as one of *metric labeling* (Kleinberg and Tardos, 2002), where label relations are encoded via a distance metric. This observation yields a meta-algorithm, applicable to both semi-supervised (via graph-theoretic techniques) and supervised settings, that alters a given n-ary classifier's output so that similar items tend to be assigned similar labels.

In what follows, we first demonstrate that humans can discern relatively small differences in (hidden) evaluation scores, indicating that rating inference is indeed a meaningful task. We then present three types of algorithms — one-vs-all, regression, and metric labeling — that can be distinguished by how explicitly they attempt to leverage similarity between items and between labels. Next, we consider what item similarity measure to apply, proposing one based on the *positive-sentence*

percentage. Incorporating this new measure within the metric-labeling framework is shown to often provide significant improvements over the other algorithms.

We hope that some of the insights derived here might apply to other scales for text classifcation that have been considered, such as clause-level opinion strength (Wilson, Wiebe, and Hwa, 2004); affect types like disgust (Subasic and Huettner, 2001; Liu, Lieberman, and Selker, 2003); reading level (Collins-Thompson and Callan, 2004); and urgency or criticality (Horvitz, Jacobs, and Hovel, 1999).

4.2 Problem validation and formulation

We first ran a small pilot study on human subjects in order to establish a rough idea of what a reasonable classification granularity is: if even people cannot accurately infer labels with respect to a five-star scheme with half stars, say, then we cannot expect a learning algorithm to do so. Indeed, some potential obstacles to accurate rating inference include lack of calibration (e.g., what an understated author intends as high praise may seem lukewarm), author inconsistency at assigning fine-grained ratings, and ratings not entirely supported by the text[2].

For data, we first collected Internet movie reviews in English from four authors, removing explicit rating indicators from each document's text automatically. Now, while the obvious experiment would be to ask subjects to guess the rating that a review represents, doing so would force us to specify a fixed rating-scale granularity in advance. Instead, we examined people's ability to discern *relative differences*, because by varying the rating differences represented by the test instances, we can evaluate multiple granularities in a single experiment. Specifically, at intervals

[2]For example, the critic Dennis Schwartz writes that "sometimes the review itself [indicates] the letter grade should have been higher or lower, as the review might fail to take into consideration my overall impression of the film — which I hope to capture in the grade" (http://www.sover.net/~ozus/cinema.htm).

Table 4.1: Human accuracy at determining relative positivity. Rating differences are given in "notches". Parentheses enclose the number of pairs attempted.

Rating diff.	Pooled	Subject 1	Subject 2
3 or more	100%	100% (35)	100% (15)
2 (e.g., 1 star)	83%	77% (30)	100% (11)
1 (e.g., $\frac{1}{2}$ star)	69%	65% (57)	90% (10)
0	55%	47% (15)	80% (5)

over a number of weeks, we authors (a non-native and a native speaker of English) examined pairs of reviews, attemping to determine whether the first review in each pair was (1) more positive than, (2) less positive than, or (3) as positive as the second. The texts in any particular review pair were taken from the same author to factor out the effects of cross-author divergence.

As Table 4.1 shows, both subjects performed perfectly when the rating separation was at least 3 "notches" in the original scale (we define a notch as a half star in a four- or five-star scheme and 10 points in a 100-point scheme). Interestingly, although human performance drops as rating difference decreases, even at a one-notch separation, both subjects handily outperformed the random-choice baseline of 33%. However, there was large variation in accuracy between subjects.[3]

Because of this variation, we defined two different classification regimes. From the evidence above, a **three-class** task (categories 0, 1, and 2 — essentially "neg-

[3]One contributing factor may be that the subjects viewed disjoint document sets, since we wanted to maximize experimental coverage of the types of document pairs within each difference class. We thus cannot report inter-annotator agreement, but since our goal is to recover a reviewer's "true" recommendation, reader-author agreement is more relevant.

While another factor might be degree of English fluency, in an informal experiment (six subjects viewing the same three pairs), native English speakers made the only two errors.

ative", "middling", and "positive", respectively) seems like one that most people would do quite well at (but we should not assume 100% human accuracy: according to our one-notch results, people may misclassify borderline cases like 2.5 stars). Our study also suggests that people could do at least fairly well at distinguishing full stars in a zero- to four-star scheme. However, when we began to construct five-category datasets for each of our four authors (see below), we found that in each case, either the most negative or the most positive class (but not both) contained only about 5% of the documents. To make the classes more balanced, we folded these minority classes into the adjacent class, thus arriving at a **four-class** problem (categories 0-3, increasing in positivity). Note that the four-class problem seems to offer more possibilities for leveraging class relationship information than the three-class setting, since it involves more class pairs. Also, as we discussed in Chapter 3, even the two-category version of the rating-inference problem for movie reviews has proven quite challenging for many automated classification techniques.

We applied the above two labeling schemes to a **scale dataset**[4] containing four corpora of movie reviews. All reviews were automatically pre-processed to remove both explicit rating indicators and objective sentences; the motivation for the latter step is that it has previously aided positive vs. negative classification (as mentioned in Section 3.3). All of the 1770, 902, 1307, or 1027 documents in a given corpus were written by the same author. This decision facilitates interpretation of the results, since it factors out the effects of different choices of methods for calibrating authors' scales.[5] We point out that it is possible to gather author-

[4]Available at http://www.cs.cornell.edu/People/pabo/movie-review-data as scale dataset v1.0.

[5]From the Rotten Tomatoes website's FAQ: "star systems are not consistent between critics. For critics like Roger Ebert and James Berardinelli, 2.5 stars or lower out of 4 stars is always negative. For other critics, 2.5 stars can either be positive or negative. Even though Eric Lurio uses a 5 star system, his grading is very relaxed. So, 2 stars can be positive." Thus, calibration may sometimes require strong familiarity with the authors involved, as anyone who has ever needed to reconcile conflicting referee reports probably knows.

specific information in some practical applications: for instance, systems that use selected authors (e.g., the Rotten Tomatoes movie-review website — where, we note, not all authors provide explicit ratings) could require that someone submit rating-labeled samples of newly-admitted authors' work. Moreover, our results at least partially generalize to mixed-author situations (see Section 4.5.2).

4.3 Algorithms

Recall that the problem we are considering is multi-category classification in which the labels can be naturally mapped to a metric space (e.g., points on a line); for simplicity, we assume the distance metric $d(\ell, \ell') = |\ell - \ell'|$ throughout. In this section, we present three approaches to this problem in order of increasingly explicit use of pairwise similarity information between items and between labels. In order to make comparisons between these methods meaningful, we base all three of them on Support Vector Machines (SVMs) as implemented in Joachims' (1999) SVM^{light} package.

4.3.1 One-vs-all

The standard SVM formulation applies only to binary classification. *One-vs-all* (OVA) (Rifkin and Klautau, 2004) is a common extension to the n-ary case. Training consists of building, for each label ℓ, an SVM binary classifier distinguishing label ℓ from "not-ℓ". We consider the final output to be a label preference function $\pi^{\text{ova}}(x, \ell)$, defined as the signed distance of (test) item x to the ℓ side of the ℓ vs. not-ℓ decision plane.

Clearly, OVA makes no explicit use of pairwise label or item relationships. However, it can perform well if each class exhibits sufficiently distinct language;

see Section 4.4 for more discussion.

4.3.2 Regression

Alternatively, we can take a *regression* perspective by assuming that the labels come from a discretization of a continuous function g mapping from the feature space to a metric space.[6] If we choose g from a family of sufficiently "gradual" functions, then similar items necessarily receive similar labels. In particular, we consider *linear, ε-insensitive* SVM regression (Vapnik, 1995; Smola and Schölkopf, 1998); the idea is to find the hyperplane that best fits the training data, but where training points whose labels are within distance ε of the hyperplane incur no loss. Then, for (test) instance x, the label preference function $\pi^{\text{reg}}(x, \ell)$ is the negative of the distance between ℓ and the value predicted for x by the fitted hyperplane function.

Wilson, Wiebe, and Hwa (2004) used SVM regression to classify clause-level strength of opinion, reporting that it provided lower accuracy than other methods. However, independently of our work, Koppel and Schler (2005) found that applying linear regression to classify documents (in a different corpus than ours) with respect to a three-point rating scale provided greater accuracy than OVA SVMs and other algorithms.

4.3.3 Metric labeling

Regression *implicitly* encodes the "similar items, similar labels" heuristic, in that one can restrict consideration to "gradual" functions. But we can also think of our task as a *metric labeling* problem (Kleinberg and Tardos, 2002), a special case of the maximum *a posteriori* estimation problem for Markov random fields, to

[6] We discuss the *ordinal* regression variant in Section 4.6.

explicitly encode our desideratum. Suppose we have an initial label preference function $\pi(x, \ell)$, perhaps computed via one of the two methods described above. Also, let d be a distance metric on labels, and let $nn_k(x)$ denote the k nearest neighbors of item x according to some item-similarity function wt. Then, it is quite natural to pose our problem as finding a mapping of instances x to labels ℓ_x (respecting the original labels of the training instances) that minimizes

$$\sum_{x \in \text{test}} \left[-\pi(x, \ell_x) + \alpha \sum_{y \in nn_k(x)} f(d(\ell_x, \ell_y)) wt(x, y) \right],$$

where f is monotonically increasing (we chose $f(d) = d$ unless otherwise specified) and α is a trade-off and/or scaling parameter. (The inner summation is familiar from work in *locally-weighted learning*[7] (Atkeson, Moore, and Schaal, 1997).) In a sense, we are using explicit item and label similarity information to increasingly penalize the initial classifier as it assigns more divergent labels to similar items.

In this chapter, we only report supervised-learning experiments in which the nearest neighbors for any given test item were drawn from the training set alone. In such a setting, the labeling decisions for different test items are independent, so that solving the requisite optimization problem is simple.

Aside: transduction The above formulation also allows for *transductive* semi-supervised learning as well, in that we could allow nearest neighbors to come from both the training and test sets. It is useful to address this case, since there are important settings in which one has a small number of labeled reviews and a large number of unlabeled reviews, in which case considering similarities between unlabeled texts could prove quite helpful. In full generality, the corresponding

[7] If we ignore the $\pi(x, \ell)$ term, different choices of f correspond to different versions of nearest-neighbor learning, e.g., majority-vote, weighted average of labels, or weighted median of labels.

multi-label optimization problem is intractable, but for many families of f functions (e.g., convex) there exist practical exact or approximation algorithms based on techniques for finding *minimum s-t cuts* in graphs (Ishikawa and Geiger, 1998; Boykov, Veksler, and Zabih, 1999; Ishikawa, 2003). As we discussed in Section 3.3, a minimum-cut formulation for the binary subjective/objective distinction yielded good results Of course, there are many other related semi-supervised learning algorithms that can be relevant; see Zhu (2005b) for a survey. In particular, as we mentioned in Section 2.1, Goldberg and Zhu (2006) presents a graph-based algorithm that addresses the semi-supervised learning setting of this problem.

4.4 Class struggle: finding a label-correlated item-similarity function

We need to specify an item similarity function wt to use the metric-labeling formulation described in Section 4.3.3. We could, as is commonly done, employ a term-overlap-based measure such as the cosine between term-frequency-based document vectors (henceforth "TO(cos)"). However, Table 4.2 shows that in aggregate, the vocabularies of distant classes overlap to a degree surprisingly similar to that of the vocabularies of nearby classes. Thus, item similarity as measured by TO(cos) may not correlate well with similarity of the item's true labels.

We can potentially develop a more useful similarity metric by asking ourselves what, intuitively, accounts for the label relationships that we seek to exploit. A simple hypothesis is that ratings can be determined by the *positive-sentence percentage (PSP)* of a text, i.e., the number of positive sentences divided by the number of subjective sentences. (As mentioned before, term-based versions of this premise have motivated much sentiment-analysis work for over a decade (Das and

Table 4.2: Average over authors and class pairs of between-class vocabulary overlap as the class labels of the pair grow farther apart.

	Label difference:		
	1	2	3
Three-class data	37%	33%	—
Four-class data	34%	31%	30%

Chen, 2001; Tong, 2001; Turney, 2002).) But counterexamples are easy to construct: reviews can contain off-topic opinions, or recount many positive aspects before describing a fatal flaw.

We therefore tested the hypothesis as follows. To avoid the need to hand-label sentences as positive or negative, we first created a *sentence polarity dataset*[8] consisting of 10,662 movie-review "snippets" (a striking extract usually one sentence long) downloaded from www.rottentomatoes.com; each snippet was labeled with its source review's label (positive or negative) as provided by Rotten Tomatoes. Then, we trained a Naive Bayes classifier on this data set and applied it to our scale dataset to identify the positive sentences (recall that objective sentences were already removed).

Figure 4.1 shows that all four authors tend to exhibit a higher PSP when they write a more positive review, and we expect that most typical reviewers would follow suit. Hence, PSP appears to be a promising basis for computing document similarity for our rating-inference task. In particular, we defined $\overrightarrow{\mathrm{PSP}(x)}$ to be the two-dimensional vector $(\mathrm{PSP}(x), 1 - \mathrm{PSP}(x))$, and then set the item-similarity

[8] Available at http://www.cs.cornell.edu/People/pabo/movie-review-data as sentence polarity dataset v1.0.

function required by the metric-labeling optimization function (Section 4.3.3) to $wt(x, y) = \cos\left(\overrightarrow{\text{PSP}(x)}, \overrightarrow{\text{PSP}(y)}\right)$.[9]

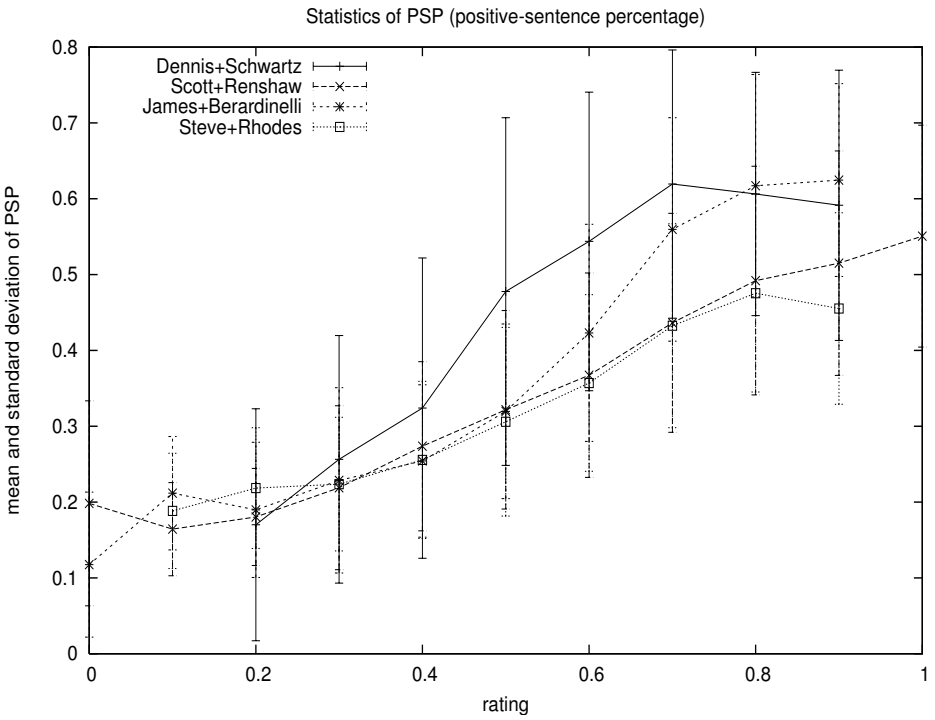


Figure 4.1: Average and standard deviation of PSP for reviews expressing different ratings.

But before proceeding, we note that it is possible that similarity information might yield no extra benefit at all. For instance, we don't need it if we can reliably identify each class just from some set of distinguishing terms. If we define such terms as frequent ones ($n \geq 20$) that appear in a single class 50% or more of the time, then we do find many instances; some examples for one author are: "meaningless", "disgusting" (class 0); "pleasant", "uneven" (class 1); and "oscar", "gem" (class 2) for the three-class case, and, in the four-class case, "flat", "tedious" (class

[9]While admittedly we initially chose this function because it was convenient to work with cosines, *post hoc* analysis revealed that the corresponding metric space "stretched" certain distances in a useful way.

1) versus "straightforward", "likeable" (class 2). Some unexpected distinguishing terms for this author are "lion" for class 2 (three-class case), and for class 2 in the four-class case, "jennifer", for a wide variety of Jennifers.

4.5 Evaluation

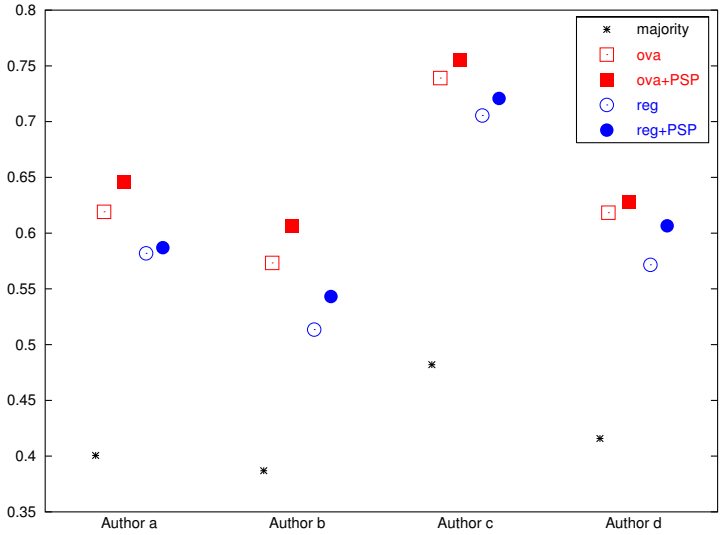
This section compares the accuracies of the approaches outlined in Section 4.3 on the four corpora comprising our scale dataset. (Results using L_1 error were qualitatively similar.) Throughout, when we refer to something as "significant", we mean statistically so with respect to the paired t-test, $p < .05$.

The results that follow are based on SVM^{light}'s default parameter settings for SVM regression and OVA. Preliminary analysis of the effect of varying the regression parameter ε in the four-class case revealed that the default value was often optimal.

The notation "A+B" denotes metric labeling where method A provides the initial label preference function π and B serves as similarity measure. To train, we first select the meta-parameters k and α by running 9-fold cross-validation within the training set. Fixing k and α to those values yielding the best performance, we then re-train A (but with SVM parameters fixed, as described above) on the whole training set. At test time, the nearest neighbors of each item are also taken from the full training set.

4.5.1 Main comparison

Figures 4.2 and 4.3 summarizes our average 10-fold cross-validation accuracy results. We first observe from the plots that all the algorithms described in Section 4.3 always definitively outperform the simple baseline of predicting the majority

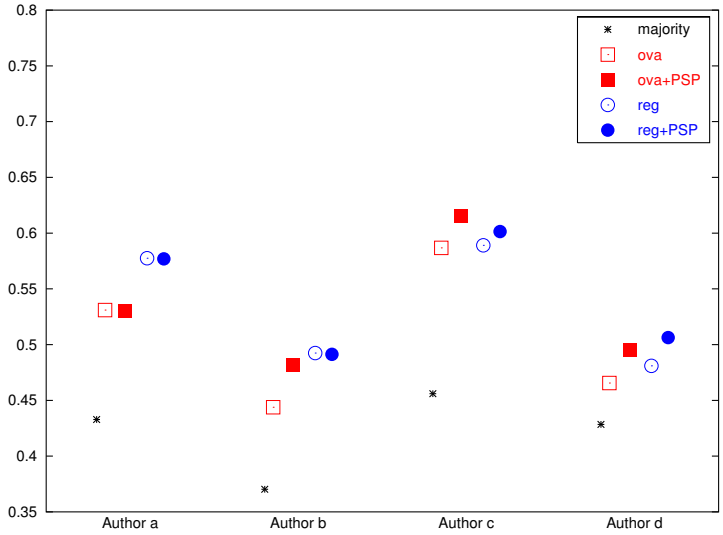


Average ten-fold cross-validation accuracies. Open icons: SVMs in either one-versus-all (square) or regression (circle) mode; dark versions: metric labeling using the corresponding SVM together with the positive-sentence percentage (PSP).

	ova				ova+PSP				reg				reg+PSP			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
ova							·		◁	◁	◁	◁	·	◁	·	·
ova+PSP	◀	◀	◀	·					◁	◁	◁	◁	◁	◁	◁	·
reg													·		·	
reg+PSP	·		·	·			·		·		·	◀		◀		

Triangles point towards significantly better algorithms for the results plotted above. Specifically, if the difference between a row and a column algorithm for a given author dataset (a, b, c, or d) is significant, a triangle points to the better one; otherwise, a dot (.) is shown. Dark icons highlight the effect of adding PSP information via metric labeling.

Figure 4.2: Results for the **three-class** classification task.



Average ten-fold cross-validation accuracies. Open icons: SVMs in either one-versus-all (square) or regression (circle) mode; dark versions: metric labeling using the corresponding SVM together with the positive-sentence percentage (PSP).

	ova	ova+PSP	reg	reg+PSP
	a b c d	a b c d	a b c d	a b c d
ova		·	· ·	· ·
ova+PSP	· ◀◀◀		· · ·	· · ·
reg	◁◁ · ·	◁ · · ·		· · · ·
reg+PSP	◁ · · ◁	◁ · · ·	• • • •	

Triangles point towards significantly better algorithms for the results plotted above. Specifically, if the difference between a row and a column algorithm for a given author dataset (a, b, c, or d) is significant, a triangle points to the better one; otherwise, a dot (.) is shown. Dark icons highlight the effect of adding PSP information via metric labeling.

Figure 4.3: Results for the **four-class** classification task.

class, although the improvements are smaller in the four-class case. Incidentally, the data was distributed in such a way that the absolute performance of the baseline itself does not change much between the three- and four-class case (which implies that the three-class datasets were relatively more balanced); and Author c's datasets seem noticeably easier than the others.

We now examine the effect of implicitly using label and item similarity. In the four-class case, regression performed better than OVA (significantly so for two authors, as shown in the righthand table); but for the three-category task, OVA significantly outperforms regression for all four authors. One might initially interprete this "flip" as showing that in the four-class scenario, item and label similarities provide a richer source of information relative to class-specific characteristics, especially since for the non-majority classes there is less data available; whereas in the three-class setting the categories are better modeled as quite distinct entities.

However, the three-class results for metric labeling on top of OVA and regression (shown in Figure 4.2 by black versions of the corresponding icons) show that employing explicit similarities always improves results, often to a significant degree, and yields the best overall accuracies. Thus, we *can* in fact effectively exploit similarities in the three-class case. Additionally, in both the three- and four- class scenarios, metric labeling often brings the performance of the weaker base method up to that of the stronger one (as indicated by the "disappearance" of upward triangles in corresponding table rows), and never hurts performance significantly.

In the four-class case, metric labeling and regression seem roughly equivalent. One possible interpretation is that the relevant structure of the problem is already captured by linear regression (and perhaps a different kernel for regression would have improved its three-class performance). However, according to additional ex-

periments we ran in the four-class situation, the test-set-optimal parameter settings for metric labeling would have produced significant improvements, indicating there may be greater potential for our framework. At any rate, we view the fact that metric labeling performed quite well for both rating scales as a definitely positive result.

4.5.2 Further discussion

Q: Metric labeling looks like it's just combining SVMs with nearest neighbors, and classifier combination often improves performance. Couldn't we get the same kind of results by combining SVMs with any other reasonable method?

A: No. For example, if we take the strongest base SVM method for initial label preferences, but replace PSP with the term-overlap-based cosine (TO(cos)), performance often drops significantly. This result, which is in accordance with Section 4.4's data, suggests that choosing an item similarity function that correlates well with label similarity is important. (ova+PSP $\lhd\lhd\lhd\lhd$ ova+TO(cos) [3c]; reg+PSP $\lhd$ reg+TO(cos) [4c])

Q: Could you explain that notation, please?

A: Triangles point toward the significantly better algorithm for some dataset. For instance, "M $\lhd\lhd\rhd$ N [3c]" means, "In the 3-class task, method M is significantly better than N for two author datasets and significantly worse for one dataset (so the algorithms were statistically indistinguishable on the remaining dataset)". When the algorithms being compared are statistically indistinguishable on all four datasets (the "no triangles" case), we indicate this with an equals sign ("=").

Q: Thanks. Doesn't Figure 4.1 show that the positive-sentence percentage would be a good classifier even in isolation, so metric labeling isn't necessary?

A: No. Predicting class labels directly from the PSP value via trained thresholds isn't as effective (ova+PSP $\lhd\lhd\lhd\lhd$ threshold PSP [3c]; reg+PSP $\lhd\lhd$ threshold PSP [4c]).

Alternatively, we could use only the PSP component of metric labeling by setting the label preference function to the constant function 0, but even with *test-set-optimal* parameter settings, doing so underperforms the *trained* metric labeling algorithm with access to an initial SVM classifier (ova+PSP $\lhd\lhd\lhd\lhd$ 0+PSP* [3c]; reg+PSP $\lhd\lhd$ 0+PSP* [4c]).

Q: What about using PSP as one of the features for input to a standard classifier?

A: Our focus is on investigating the utility of similarity information. In our particular rating-inference setting, it so happens that the basis for our pairwise similarity measure can be incorporated as an item-specific feature, but we view this as a tangential issue. That being said, preliminary experiments show that metric labeling can be better, barely (for test-set-optimal parameter settings for both algorithms: significantly better results for one author, four-class case; statistically indistinguishable otherwise), although one needs to determine an appropriate weight for the PSP feature to get good performance.

Q: You defined the "metric transformation" function f as the identity function $f(d) = d$, imposing greater loss as the distance between labels assigned to two similar items increases. Can you do just as well if you penalize all non-equal label assignments by the same amount, or does the distance between labels really matter?

A: You're asking for a comparison to the *Potts model*, which sets f to the function $\hat{f}(d) = 1$ if $d > 0$, 0 otherwise. In the one setting in which there is a significant difference between the two, the Potts model does worse (ova+PSP $\lhd$ ova$\hat{+}$PSP [3c]).

Also, employing the Potts model generally leads to fewer significant improvements over a chosen base method (compare Figures 4.2 and 4.3's tables with: reg$\hat{+}$PSP $\lhd$ reg [3c]; ova$\hat{+}$PSP $\lhd\lhd$ ova [3c]; ova$\hat{+}$PSP = ova [4c]; but note that reg$\hat{+}$PSP $\lhd$ reg [4c]). We note that optimizing the Potts model in the multi-label case is NP-hard, whereas the optimal metric labeling with the identity metric-transformation function can be efficiently obtained (see Section 4.3.3).

Q: Your datasets had many labeled reviews and only one author each. Is your work relevant to settings with many authors but very little data for each?

A: As discussed in Section 4.2, it can be quite difficult to properly calibrate different authors' scales, since the same number of "stars" even within what is ostensibly the same rating system can mean different things for different authors. But since you ask: we temporarily turned a blind eye to this serious issue, creating a collection of 5394 reviews by 496 authors with at most 80 reviews per author, where we pretended that our rating conversions mapped correctly into a universal rating scheme. Preliminary results on this dataset were actually comparable to the results reported above, although since we are not confident in the class labels themselves, more work is needed to derive a clear analysis of this setting. (Abusing notation, since we're already playing fast and loose: [3c]: baseline 52.4%, reg 61.4%, reg+PSP 61.5%, ova (65.4%) $\rhd$ ova+PSP (66.3%); [4c]: baseline 38.8%, reg (51.9%) $\rhd$ reg+PSP (52.7%), ova (53.8%) $\rhd$ ova+PSP (54.6%))

In future work, it would be interesting to determine author-independent characteristics that can be used on (or suitably adapted to) data for specific authors.

Q: How about trying —

A: —Yes, there are many alternatives. A few that we tested are described in

Section 4.7, and we propose some others in the next section. We should mention that we have not yet experimented with *all-vs.-all* (AVA), another standard binary-to-multi-category classifier conversion method, because we wished to focus on the effect of omitting pairwise information. In independent work on 3-category rating inference for a different corpus, Koppel and Schler (2005) found that regression outperformed AVA, and Rifkin and Klautau (2004) argue that in principle OVA should do just as well as AVA.

4.6 Related work and future directions

In this chapter, we addressed the rating-inference problem, showing the utility of employing label similarity and (appropriate choice of) item similarity — either implicitly, through regression, or explicitly and often more effectively, through metric labeling.

In the future, we would like to apply our methods to other scale-based classification problems, and explore alternative methods. Clearly, varying the kernel in SVM regression might yield better results. Another choice is *ordinal regression* (McCullagh, 1980; Herbrich, Graepel, and Obermayer, 2000), which only considers the ordering on labels, rather than any explicit distances between them; this approach could work well if a good metric on labels is lacking. Also, one could use mixture models (e.g., combine "positive" and "negative" language models) to capture class relationships (McCallum, 1999; Schapire and Singer, 2000; Takamura, Matsumoto, and Yamada, 2004).

We are also interested in framing multi-class but *non*-scale-based categorization problems as metric labeling tasks. For example, positive vs. negative vs. neutral sentiment distinctions are sometimes considered in which neutral means either

objective (Engström, 2004) or a conflation of objective with a rating of mediocre (Das and Chen, 2001). (Koppel and Schler (2005) in independent work also discuss various types of neutrality.) In either case, we could apply a metric in which positive and negative are closer to objective (or objective+mediocre) than to each other. As another example, hierarchical label relationships can be easily encoded in a label metric.

Finally, as mentioned in Section 4.3.3, Goldberg and Zhu (2006) address the transductive setting, in which one has a small amount of labeled data and uses relationships between unlabeled items; this setting would also be well-suited to the metric-labeling approach and may be interesting to try out in the future.

4.7 Other variations attempted

4.7.1 Discretizing binary classification

In our setting, we can also incorporate class relations by directly altering the output of a binary classifier, as follows. We first train a standard SVM, treating ratings greater than 0.5 as positive labels and others as negative labels. If we then consider the resulting classifier to output a *positivity-preference function* $\pi_+(x)$, we can then learn a series of thresholds to convert this value into the desired label set, under the assumption that the bigger $\pi_+(x)$ is, the more positive the review.[10] This algorithm always outperforms the majority-class baseline, but not to the degree that the best of SVM OVA and SVM regression does. Koppel and Schler (2005) independently found in a three-class study that thresholding a positive/negative classifier trained only on clearly positive or clearly negative examples did not yield

[10]This is not necessarily true: if the classifier's goal is to optimize binary classification error, its major concern is to increase confidence in the positive/negative distinction, which may not correspond to higher confidence in separating "five stars" from "four stars".

large improvements.

4.7.2 Discretizing regression

In our experiments with SVM regression, we discretized regression output via a set of fixed decision thresholds $\{0.5, 1.5, 2.5, ...\}$ to map it into our set of class labels. Alternatively, we can learn the thresholds instead. Neither option clearly outperforms the other in the four-class case. In the three-class setting, the learned version provides noticeably better performance in two of the four datasets. But these results taken together still mean that in many cases, the difference is negligible, and if we had started down this path, we would have needed to consider similar tweaks for one-vs-all SVM as well. We therefore stuck with the simpler version in order to maintain focus on the central issues at hand.

Chapter 5

Variation on the theme of polarity and document relationship with politically oriented text[1]

5.1 Introduction: determining support or opposition from Congressional floor-debate transcripts

> *One ought to recognize that the present political chaos is connected with the decay of language, and that one can probably bring about some improvement by starting at the verbal end.*
>
> — Orwell, *Politics and the English language*

We have entered an era where very large amounts of politically oriented text are now available online. This includes both official documents, such as the full text of laws and the proceedings of legislative bodies, and unofficial documents, such as

[1]This chapter is based on the paper "Get out the vote: Determining support or opposition from congressional floor-debate transcripts" with Matt Thomas and Lillian Lee, which will appear in the proceedings of EMNLP 2006 (Thomas, Pang, and Lee, 2006).

postings on weblogs (blogs) devoted to politics. In some sense, the availability of such data is simply a manifestation of a general trend of "everybody putting their records on the Internet".[2] The online accessibility of politically oriented texts in particular, however, is a phenomenon that some have gone so far as to say will have a potentially society-changing effect.

In the United States, for example, governmental bodies are providing and soliciting political documents via the Internet, with lofty goals in mind: *electronic rulemaking* (eRulemaking) initiatives involving the "electronic collection, distribution, synthesis, and analysis of public commentary in the regulatory rulemaking process", may "[alter] the citizen-government relationship" (Shulman and Schlosberg, 2002). Additionally, much media attention has been focused recently on the potential impact that Internet sites may have on politics[3], or at least on political journalism[4]. Regardless of whether one views such claims as clear-sighted prophecy or mere hype, it is obviously important to help people understand and analyze politically oriented text, given the importance of enabling informed participation in the political process.

Evaluative and persuasive documents, such as a politician's speech regarding a bill or a blogger's commentary on a legislative proposal, form a particularly interesting type of politically oriented text. People are much more likely to consult such evaluative statements than the actual text of a bill or law under discussion, given the dense nature of legislative language and the fact that (U.S.) bills often reach several hundred pages in length (Smith, Roberts, and Vander Wielen, 2005).

[2]It is worth pointing out that the United States' Library of Congress was an extremely early adopter of Web technology: the THOMAS database (http://thomas.loc.gov) of congressional bills and related data was launched in January 1995, when Mosaic was not quite two years old and Altavista did not yet exist.

[3]E.g., "Internet injects sweeping change into U.S. politics", Adam Nagourney, *The New York Times*, April 2, 2006.

[4]E.g., "The End of News?", Michael Massing, *The New York Review of Books*, December 1, 2005.

Moreover, political opinions are explicitly solicited in the eRulemaking scenario.

In the analysis of evaluative language, it is fundamentally necessary to determine whether the author/speaker supports or disapproves of the topic of discussion. In this chapter, we investigate the following specific instantiation of this problem: we seek to determine from the transcripts of U.S. Congressional floor debates whether each "speech" (continuous single-speaker segment of text) represents support for or opposition to a proposed piece of legislation. Note that from an experimental point of view, this is a very convenient problem to work with because we can automatically determine ground truth (and thus avoid the need for manual annotation) simply by consulting publicly available voting records.

Task properties Determining whether or not a speaker supports a proposal again falls within the realm of sentiment analysis. In particular, since we treat each individual speech within a debate as a single "document", we are considering a version of sentiment-polarity classification discussed in Chapter 3.

Most sentiment-polarity classifiers proposed in the recent literature categorize each document independently. A few others, including our own work described in Chapter 4 incorporate various measures of inter-document similarity between the texts to be labeled (Agarwal and Bhattacharyya, 2005; Pang and Lee, 2005; Goldberg and Zhu, 2006). Many interesting opinion-oriented documents, however, can be linked through certain relationships that occur in the context of evaluative *discussions*. For example, we may find textual[5] evidence of a high likelihood of

[5]Because we are most interested in techniques applicable across domains, we restrict consideration to NLP aspects of the problem, ignoring external problem-specific information. For example, although most votes in our corpus were almost completely along party lines (and despite the fact that same-party information is easily incorporated via the methods we propose), we did not use party-affiliation data. Indeed, in other settings (e.g., a movie-discussion listserv) one may not be able to determine the participants' political leanings, and such information may not lead to significantly improved results even if it were available.

agreement between two speakers, such as explicit assertions ("I second that!") or quotation of messages in emails or postings (see Mullen and Malouf (2006) but cf. Agrawal et al. (2003), who observed responses in newsgroups are more likely to antagonistic than reinforcing). Agreement evidence can be a powerful aid in our classification task: for example, we can easily categorize a complicated (or overly terse) document if we find within it indications of agreement with a clearly positive text.

Obviously, incorporating agreement information provides additional benefit only when the input documents are relatively difficult to classify individually. Intuition suggests that this is true of the data with which we experiment, for several reasons. First, U.S. congressional debates contain very rich language and cover an extremely wide variety of topics, ranging from flag burning to international policy to the federal budget. Debates are also subject to digressions, some fairly natural and others less so (e.g., "Why are we discussing this bill when the plight of my constituents regarding this other issue is being ignored?")

Second, an important characteristic of persuasive language is that speakers may spend more time presenting evidence in support of their positions (or attacking the evidence presented by others) than directly stating their attitudes. An extreme example will illustrate the problems involved. Consider a speech that describes the U.S. flag as deeply inspirational, and thus contains only positive language. If the bill under discussion is a proposed flag-burning ban, then the speech is *supportive*; but if the bill under discussion is aimed at rescinding an existing flag-burning ban, the speech may represent *opposition* to the legislation. Given the current state of the art in sentiment analysis, it is doubtful that one could determine the (probably topic-specific) relationship between presented evidence and speaker opinion.

Table 5.1: Corpus statistics.

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

Qualitative summary of results The above difficulties underscore the importance of enhancing standard classification techniques with new information sources that promise to improve accuracy, such as inter-document relationships between the documents to be labeled. In this work, we demonstrate that the incorporation of agreement modeling can provide substantial improvements over the application of support vector machines (SVMs) in isolation, which represents the state of the art in the individual classification of documents. The enhanced accuracies are obtained via a fairly primitive automatically-acquired "agreement detector" and a conceptually simple method for integrating isolated-document and agreement-based information. We thus view our results as demonstrating the potentially large benefits of exploiting sentiment-related discourse-segment relationships in sentiment-analysis tasks.

5.2 Corpus: transcripts of U.S. floor debates

This section outlines the main steps of the process by which we created our corpus (download site: www.cs.cornell.edu/home/llee/data/convote.html).

GovTrack (http://govtrack.us) is an independent website run by Joshua Tauberer

that collects publicly available data on the legislative and fund-raising activities of U.S. congresspeople. Due to its extensive cross-referencing and collating of information, it was nominated for a 2006 "Webby" award. A crucial characteristic of GovTrack from our point of view is that the information is provided in a very convenient format; for instance, the floor-debate transcripts are broken into separate HTML files according to the subject of the debate, so we can trivially derive long sequences of speeches guaranteed to cover the same topic.

We extracted from GovTrack all available transcripts of U.S. floor debates in the House of Representatives for the year 2005 (3268 pages of transcripts in total), together with voting records for all roll-call votes during that year. We concentrated on debates regarding "controversial" bills (ones in which the losing side generated at least 20% of the speeches) because these debates should presumably exhibit more interesting discourse structure.

Each debate consists of a series of *speech segments*, where each segment is a sequence of uninterrupted utterances by a single speaker. Since speech segments represent natural discourse units, we treat them as the basic unit to be classified. Each speech segment was labeled by the vote ("yea" or "nay") cast for the proposed bill by the person who uttered the speech segment.

We automatically discarded those speech segments belonging to a class of formulaic, generally one-sentence utterances focused on the yielding of time on the house floor (for example, "Madam Speaker, I am pleased to yield 5 minutes to the gentleman from Massachusetts"), as such speech segments are clearly off-topic. We also removed speech segments containing the term "amendment", since we found during initial inspection that these speeches generally reflect a speaker's opinion on an amendment, and this opinion may differ from the speaker's opinion on the

underlying bill under discussion.

We randomly split the data into training, test, and development (parameter-tuning) sets representing roughly 70%, 20%, and 10% of our data, respectively (see Table 5.1). The speech segments remained grouped by debate, with 38 debates assigned to the training set, 10 to the test set, and 5 to the development set; we require that the speech segments from an individual debate all appear in the same set because our goal is to examine classification of speech segments in the context of the surrounding discussion.

5.3 Method

The support/oppose classification problem can be approached through the use of standard classifiers such as support vector machines (SVMs), which consider each text unit in isolation. As discussed in Section 5.1, however, the conversational nature of our data implies the existence of various relationships that can be exploited to improve cumulative classification accuracy for speech segments belonging to the same debate. We adopt the same graph-cut formulation described in Section 3.3 as our classification framework, which integrates both perspectives, optimizing its labeling of speech segments based on both individual speech-segment classification scores and preferences for groups of speech segments to receive the same label.

More specifically, let $s_1, s_2, \ldots, s_n$ be the sequence of speech segments within a given debate, and let $\mathcal{Y}$ and $\mathcal{N}$ stand for the "yea" and "nay" class, respectively. Assume we have a non-negative function $ind(s, C)$ indicating the degree of preference that an individual-document classifier, such as an SVM, has for placing speech-segment s in class C. Also, assume that some pairs of speech segments have *weighted links* between them, where the non-negative *strength* (weight) $str(\ell)$ for a

link ℓ indicates the degree to which it is preferable that the linked speech segments receive the same label. Then, any class assignment $c = c(s_1), c(s_2), \ldots, c(s_n)$ can be assigned a *cost*

$$\sum_{s} ind(s, \overline{c}(s)) + \sum_{s,s' \colon c(s) \neq c(s')} \sum_{\ell \text{ between } s,s'} str(\ell),$$

where $\overline{c}(s)$ is the "opposite" class from $c(s)$. Again, a *minimum-cost* assignment represents an optimum way to classify the speech segments so that each one tends not to be put into the class that the individual-document classifier disprefers, but at the same time, highly associated speech segments tend not to be put in different classes.

The above optimization function and the advantage of using this framework has been discussed in Section 3.3. The contribution of our work in this chapter is the examination of new types of relationships, not the method by which such relationships are incorporated into the classification decision.

5.3.1 Classifying speech segments in isolation

In our experiments, we employed the well-known classifier SVM^{light} to obtain individual-document classification scores, treating $\mathcal{Y}$ as the positive class and using plain unigrams as features.[6] Following standard practice in sentiment analysis (Pang, Lee, and Vaithyanathan, 2002), the input to SVM^{light} consisted of normalized presence-of-feature (rather than frequency-of-feature) vectors[7]. The *ind* value for each speech segment s was based on the signed distance $d(s)$ from the vector

[6] SVM^{light} is available at svmlight.joachims.org. Default parameters were used, although experimentation with different parameter settings is an important direction for future work (Daelemans and Hoste, 2002; Munson, Cardie, and Caruana, 2005).

[7] Note that this was in fact suboptimal in some settings for this particular dataset

representing s to the trained SVM decision plane:

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

where σ_s is the standard deviation of $d(s)$ over all speech segments s in the debate in question, and $ind(s, \mathcal{N}) \stackrel{\text{def}}{=} 1 - ind(s, \mathcal{Y})$. This way, any $d(s)$ between $-2\sigma_s$ and $2\sigma_s$ (for normally distributed data, this should cover 95% of the population) is normalized into an *ind* score between 0 and 1.

We now turn to the more interesting problem of representing the preferences that speech segments may have for being assigned to the same class.

5.3.2 Relationships between speech segments

A wide range of relationships between text segments can be modeled as positive-strength links. Here we discuss two types of constraints that are considered in this work.

Same-speaker constraints: In Congressional debates and in general social-discourse contexts, a single speaker may make a number of comments regarding a topic. It is reasonable to expect that in many settings, the participants in a discussion may be convinced to change their opinions midway through a debate. Hence, in the general case we wish to be able to express "soft" preferences for all of an author's statements to receive the same label, where the strengths of such constraints could, for instance, vary according to the time elapsed between the statements. Weighted links are an appropriate means to express such variation.

However, if we assume that most speakers do not change their positions in the course of a discussion, we can conclude that all comments made by the same speaker must receive the same label. This assumption holds by fiat for the ground-truth labels in our dataset because these labels were derived from the single vote cast by the speaker on the bill being discussed.[8] We can implement this assumption via links whose weights are essentially infinite. Although one can also implement this assumption via concatenation of same-speaker speech segments (see Section 5.4.3), we view the fact that our graph-based framework incorporates both hard and soft constraints in a principled fashion as an advantage of our approach.

Different-speaker agreements In House discourse, it is common for one speaker to make reference to another in the context of an agreement or disagreement over the topic of discussion. The systematic identification of instances of agreement can, as we have discussed, be a powerful tool for the development of intelligently selected weights for links between speech segments.

The problem of agreement identification can be decomposed into two sub-problems: identifying references and their targets, and deciding whether each reference represents an instance of agreement. In our case, the first task is straightforward because we focused solely on by-name references. Members of Congress typically refer to each other by formulaic structures such as "the gentleman from Ohio", and House transcripts accompany each such reference with the last name of the reference target (e.g. "Mr. Smith") in parentheses. To isolate references in our implementation, we examine all statements of names beginning with Mr.,

[8] We are attempting to determine whether a speech segment represents support or not. This differs from the problem of determining what the speaker's actual opinion is, a problem that, as an anonymous reviewer put it, is complicated by "grandstanding, backroom deals, or, more innocently, plain change of mind ('I voted for it before I voted against it')".

Ms., or Mrs.,[9] and we attempt to match each such reference with a House member using a directory obtained from Govtrack. In cases in which multiple members could potentially match a reference based on a common last name, we give priority to members who spoke in close proximity to the speech segment containing the reference.

Our second, and more interesting, task with regard to reference processing is to decide whether a given reference indicates agreement. We approach the problem of classifying references by representing each reference with a word-presence vector derived from a window of text surrounding the reference.[10] In the training set, we classify each reference connecting two speakers with a positive or negative label depending on whether the two voted the same way on the bill under discussion[11]. These labels are then used to train an SVM classifier, the output of which is subsequently used to create weights on *agreement links* in the test set as follows.

Let $d(r)$ denote the distance from the vector representing reference r to the agreement-detector SVM's decision plane, and let σ_r be the standard deviation of $d(r)$ over all references in the debate in question. We then define the strength *agr* of the *agreement link* corresponding to the reference as:

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

The free parameter α specifies the relative importance of the *agr* scores. The

[9] One subtlety is that for the purposes of mining agreement cues (but *not* for evaluating overall support/oppose classification accuracy), we temporarily re-inserted into our dataset previously filtered speech segments containing the term "yield", since the yielding of time on the House floor typically indicates agreement even though the yield statements contain little relevant text on their own.

[10] We found good development-set performance using the 30 tokens before, 20 tokens after, and the name itself.

[11] Since we are concerned with references that potentially represent relationships between speech segments, we ignore references for which the target of the reference did not speak in the debate in which the reference was made.

threshold θ_{agr} controls the precision of the agreement links, in that values of θ_{agr} greater than zero mean that greater confidence is required before an agreement link can be added.[12]

5.4 Evaluation

This section presents experiments testing the utility of using speech-segment relationships, evaluating against a number of baselines. All reported results use values for the free parameter α derived via tuning on the development set. In the tables, **boldface** indicates the development- and test-set results for the *development-set-optimal* parameter settings, as one would make algorithmic choices based on development-set performance.

5.4.1 Preliminaries: Reference classification

Recall that to gather inter-speaker agreement information, the strategy employed in this work is to classify by-name references to other speakers as to whether they indicate agreement or not.

To train our agreement classifier, we experimented with undoing the deletion of amendment-related speech segments in the training set. Note that such speech segments were *never* included in the development or test set, since, as discussed in Section 5.2, their labels are probably noisy; however, including them in the *training* set allows the classifier to examine more instances even though some of them are labeled incorrectly. As Table 5.2 shows, using more, if noisy, data yields better agreement-classification results on the development set, and so we use that

[12] Our implementation puts a link between just one arbitrary pair of speech segments among all those uttered by a given pair of apparently agreeing speakers. The "infinite-weight" same-speaker links propagate the agreement information to all other such pairs.

Table 5.2: Agreement-classifier accuracy, in percent. "Amendments"="speech segments containing the word 'amendment'". Recall that boldface indicates results for development-set-optimal settings.

Agreement classifier	Devel. set	Test set
majority baseline	81.51	80.26
Train: no amendments; $\theta_{\text{agr}} = 0$	84.25	81.07
Train: with amendments; $\theta_{\text{agr}} = 0$	**86.99**	**80.10**

policy in all subsequent experiments.[13]

An important observation is that precision may be more important than accuracy in deciding which agreement links to add: false positives with respect to agreement can cause speech segments to be incorrectly assigned the same label, whereas false negatives mean only that agreement-based information about other speech segments is not employed. As described above, we can raise agreement precision by increasing the threshold θ_{agr}, which specifies the required confidence for the addition of an agreement link. Indeed, Table 5.3 shows that we can improve agreement precision by setting θ_{agr} to the (positive) mean agreement score μ assigned by the SVM agreement-classifier over all references in the given debate[14]. However, this comes at the cost of greatly reducing agreement accuracy (development: 64.38%; test: 66.18%) due to lowered recall levels. Whether or not better speech-segment classification is ultimately achieved is discussed in the next sections.

[13]Unfortunately, this policy leads to inferior *test-set* agreement classification. Section 5.4.5 contains further discussion.

[14]We elected not to explicitly tune the value of θ_{agr} in order to minimize the number of free parameters to deal with.

Table 5.3: Agreement-classifier precision.

Agreement classifier	Precision (in percent):	
	Devel. set	Test set
$\theta_{\text{agr}} = 0$	86.23	82.55
$\theta_{\text{agr}} = \mu$	**89.41**	**88.47**

Table 5.4: Segment-based speech-segment classification accuracy, in percent.

Support/oppose classifer	Devel. set	Test set
majority baseline	54.09	58.37
#("support") − #("oppos")	59.14	62.67
SVM [speech segment]	70.04	66.05
SVM + same-speaker links	79.77	67.21
SVM + same-speaker links + agreement links, $\theta_{\text{agr}} = 0$	**89.11**	**70.81**
SVM + same-speaker links + agreement links, $\theta_{\text{agr}} = \mu$	87.94	71.16

5.4.2 Segment-based speech-segment classification

Baselines The first two data rows of Table 5.4 depict baseline performance results. The #("support") − #("oppos") baseline is meant to explore whether the speech-segment classification task can be reduced to simple lexical checks. Specifically, this method uses the signed difference between the number of words containing the stem "support" and the number of words containing the stem "oppos" (returning the majority class if the difference is 0). No better than 62.67% test-set accuracy is obtained by either baseline.

Table 5.5: Speaker-based speech-segment classification accuracy, in percent. Here, the initial SVM is run on the concatenation of all of a given speaker's speech segments, but the results are computed over speech segments (not speakers), so that they can be compared to those in Table 5.4.

Support/oppose classifer	Devel. set	Test set
SVM [speaker]	71.60	70.00
SVM + agreement links, $\theta_{\text{agr}} = 0$	**88.72**	**71.28**
SVM + agreement links, $\theta_{\text{agr}} = \mu$	84.44	76.05

Using relationship information Applying an SVM to classify each speech segment in isolation leads to clear improvements over the two baseline methods, as demonstrated in Table 5.4. When we impose the constraint that all speech segments uttered by the same speaker receive the same label via "same-speaker links", both test-set and development-set accuracy increase even more, in the latter case quite substantially so.

The last two lines of Table 5.4 show that the best results are obtained by incorporating agreement information as well. The highest test-set result, 71.16%, is obtained by using a high-precision threshold to determine which agreement links to add. While the development-set results would induce us to utilize the standard threshold value of 0, which is sub-optimal on the test set, the $\theta_{\text{agr}} = 0$ agreement-link policy still achieves noticeable improvement over not using agreement links (test set: 70.81% vs. 67.21%).

5.4.3 Speaker-based speech-segment classification

We use speech segments as the unit of classification because they represent natural discourse units. As a consequence, we are able to exploit relationships at the speech-segment level. However, it is interesting to consider whether we really need to consider relationships specifically between speech segments themselves, or whether it suffices to simply consider relationships between the *speakers* of the speech segments. In particular, as an alternative to using same-speaker links, we tried a *speaker-based* approach wherein the way we determine the initial individual-document classification score for each speech segment uttered by a person p in a given debate is to run an SVM on the concatenation of *all* of p's speech segments within that debate. (We also ensure that agreement-link information is propagated from speech-segment to speaker pairs.)

How does the use of same-speaker links compare to the concatenation of each speaker's speech segments? Tables 5.4 and 5.5 show that, not surprisingly, the SVM individual-document classifier works better on the concatenated speech segments than on the speech segments in isolation. However, the effect on overall classification accuracy is less clear: the development set favors same-speaker links over concatenation, while the test set does not.

But we stress that the most important observation we can make from Table 5.5 is that once again, the addition of agreement information leads to substantial improvements in accuracy.

5.4.4 "Hard" agreement constraints

Recall that in in our experiments, we created finite-weight agreement links, so that speech segments appearing in pairs flagged by our (imperfect) agreement detector

Table 5.6: Speech-segment classification accuracy with hard agreement constraints. We force speech segments with agreement links between them to receive the same label, either through using infinite-weight for agreement links (for both segment-based formulation and speaker-based formulation) or through a concatenation strategy similar to that described in the previous subsection.

Support/oppose classifer	Setting	Devel. set	Test set
Graph with infinite-weight links	$\theta_{\text{agr}} = \mu$	80.93	64.77
Segment-based (cf. Table 5.4)	$\theta_{\text{agr}} = \mu$	84.44	66.16
Graph with infinite-weight links	$\theta_{\text{agr}} = 0$	76.26	68.26
Speaker-based (cf. Table 5.5)	$\theta_{\text{agr}} = \mu$	76.26	69.07
Concatenation	$\theta_{\text{agr}} = 0$	59.14	63.14
(cf. Table 5.4 and 5.5)	$\theta_{\text{agr}} = \mu$	82.88	64.07

can potentially receive different labels. We also experimented with *forcing* such speech segments to receive the same label, either through infinite-weight agreement links or through a speech-segment concatenation strategy similar to that described in the previous subsection. Both strategies resulted in clear degradation in performance on both the development and test sets, a finding that validates our encoding of agreement information as "soft" preferences.

5.4.5 On the development/test set split

We have seen several cases in which the method that performs best on the development set does not yield the best test-set performance. Also, in retrospect, the size of the development set may be too small. However, we felt that it would be

illegitimate to change the train/development/test sets in a post hoc fashion, that is, after seeing the experimental results.

Moreover, and crucially, it is very clear that using agreement information, encoded as preferences within our graph-based approach rather than as hard constraints, yields substantial improvements on both the development and test set; this, we believe, is our most important finding.

5.5 Related work

Politically-oriented text Sentiment analysis has specifically been proposed as a key enabling technology in eRulemaking, allowing the automatic analysis of the opinions that people submit (Shulman et al., 2005; Cardie et al., 2006; Kwon, Shulman, and Hovy, 2006). There has also been work focused upon determining the political leaning (e.g., "liberal" vs. "conservative") of a document or author, where most previously-proposed methods make no direct use of relationships between the documents to be classified (the "unlabeled" texts) (Laver, Benoit, and Garry, 2003; Efron, 2004; Mullen and Malouf, 2006). An exception is Grefenstette et al. (2004), who experimented with determining the political orientation of websites essentially by classifying the concatenation of all the documents found on that site.

Others have applied the NLP technologies of near-duplicate detection and topic-based text categorization to politically oriented text (Yang and Callan, 2005; Purpura and Hillard, 2006).

Detecting agreement We used a simple method to learn to identify cross-speaker references indicating agreement. More sophisticated approaches have been proposed (Hillard, Ostendorf, and Shriberg, 2003), including an extension that, in

an interesting reversal of our problem, makes use of sentiment-polarity indicators within speech segments (adjectives with positive or negative polarity) to help derive the agreement information (Galley et al., 2004)[15]. Also relevant is work on the general problems of dialog-act tagging (Stolcke et al., 2000), citation analysis (Lehnert, Cardie, and Riloff, 1990), and computational rhetorical analysis (Marcu, 2000; Teufel and Moens, 2002).

We currently do not have an efficient means to encode *disagreement* information as hard constraints; we plan to investigate incorporating such information in future work.

Relationships between the unlabeled items Carvalho and Cohen (2005) consider sequential relations between different types of emails (e.g., between requests and satisfactions thereof) to classify messages, and thus also explicitly exploit the structure of conversations.

Previous sentiment-analysis work in different domains has considered inter-document similarity (Agarwal and Bhattacharyya, 2005; Pang and Lee, 2005; Goldberg and Zhu, 2006) or explicit inter-document references in the form of hyperlinks (Agrawal et al., 2003).

5.6 Conclusion and future work

In this study, we focused on very general types of cross-document classification preferences, utilizing constraints based only on speaker identity and on direct textual references between statements. We showed that the integration of even very limited information regarding inter-document relationships can significantly increase

[15]Note that Galley et al. (2004) work with actual speech signals in the context of conversational interacation. They incorporated a number of non-texual features (such as utterance length) as well as manually-selected lexical items in their agreement classifier.

the accuracy of support/opposition classification.

The simple constraints modeled in our study, however, represent just a small portion of the rich network of relationships that connect statements and speakers across the political universe and in the wider realm of opinionated social discourse. One intriguing possibility is to take advantage of (readily identifiable) information regarding interpersonal relationships, making use of speaker/author affiliations, positions within a social hierarchy, and so on. Or, we could even attempt to model relationships between topics or concepts, in a kind of extension of collaborative filtering. For example, perhaps we could infer that two speakers sharing a common opinion on evolutionary biologist Richard Dawkins (a.k.a. "Darwin's rottweiler") will be likely to agree in a debate centered on Intelligent Design. While such functionality is well beyond the scope of our current study, we are optimistic that we can develop methods to exploit additional types of relationships in future work.

Chapter 6

Rhapsody on the theme of subjectivity: collective document-level subjectivity detection without training data[1]

6.1 Introduction

> *There never was in the world two opinions alike, no more than two hairs, or two grains; the most universal quality is diversity.*
>
> — Montaigne, *Essays*

> *Where an opinion is general, it is usually correct.*
>
> — Austen, *Mansfield Park*

What other people think has always been an important piece of information for most of us during the decision-making process. Long before awareness of the World

[1]This chapter is based on an earlier version of the work described in "Using very simple statistics for review search: An exploration" with Lillian Lee, which appeared in the proceedings of COLING (Companion volume: Posters) (Pang and Lee, 2008).

Wide Web became widespread, many of us asked our friends to recommend an auto mechanic, requested reference letters from colleagues, or consulted *Consumer Reports* to decide what dishwasher to buy. Interestingly, the Web has (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of.

Websites that aggregate and organize *solicited* reviews, such as epinions.com, amazon.com, and tripadvisor.com, have proven to be highly valuable sources of information. But a system that could *proactively search* the Web for reviews of a given item offers many benefits as well. Such a system could be used to augment the contents of websites like those just mentioned. Or, users could employ such a system directly, especially if it were integrated into their search engine of choice. Indeed, users might find this capability a useful alternative to consulting aggregation sites. Specifically, while the element of trust might be lost (many aggregation sites have some way to indicate the apparent quality of reviewers), what can be gained is access to potentially many more reviews from a wider variety of people (or from a variety of review aggregators), and the ability to find reviews on topics not well covered by known review sites.

It might seem that one can already use typical search engines in this manner via suitably worded queries; for instance, one can add the term "review" to the list of query keywords. But this is an imperfect solution. For instance, at the time of writing, querying Google for "Novotel Sydney review" brings up a number of relevant pages, but also pages containing versions of the dreaded "This hotel hasn't been reviewed yet. Be the first to review it".[2] Heuristics such as looking

[2]The hotel's Web policy was also retrieved: "The Hotel does not *review* or monitor any web sites linked to the Site" (emphasis added).

for a rating indication (e.g., "two stars") might help, but given the potentially wide range of site-specific rating indicators, we do not expect this to be a general solution.

Instead, one approach would be to apply document-level *subjectivity classification* to the initial set of search engine results, either to filter out non-reviews completely or simply to distinguish the subjective from the objective documents. However, although the input consists of a collection of documents more or less on the same focused topic (having been generated in response to a specific query), it is impossible to determine a priori what the set of possible topics could be or what sources the documents will be drawn from.[3] This poses a serious problem for existing techniques. As we discussed in Chapter 2, supervised sentiment-analysis

technologies have been shown to be sensitive to changes in domain or even different time-slices of the same domain (Dave, Lawrence, and Pennock, 2003; Aue and Gamon, 2005b; Finn and Kushmerick, 2006; Read, 2005). Furthermore, even some unsupervised methods, which generally rely on a set of typically subjectivity-indicative words or phrases (Wiebe et al., 2004), exhibit quite different accuracies for different domains, at least with respect to related sentiment-analysis classification problems (Turney, 2002); also, an unsupervised method relying solely on very general cues ignores potentially strong evidence that can be drawn from the topic-specific documents at hand — for this reason, Dave, Lawrence, and Pennock (2003) specifically suggest that a "specialized genre classifier" is needed for mining opinions. What we really desire, then, is a method for *unsupervised, "blank-slate" subjectivity classification*, where we assume access neither to labeled data nor to prior linguistic information regarding subjectivity indicators, but rather draw our

[3]Recall from Chapter 2, performance results for certain *within*-genre distinctions, such as separating editorials from non-opinion newswire, can be quite high (Yu and Hatzivassiloglou, 2003; Wiebe et al., 2004).

inferences almost entirely[4] from the test data at hand.

To develop such a method, we consider the following simple hypothesis: information that is repeated across the vast majority of the input documents, such as "100 Murray Street" in the case of the Novotel Sydney query, is likely to be factual, whereas opinions exhibit a higher degree of variance. If the hypothesis is true, it suggests a general strategy of classifying documents as non-reviews when their contents tend to be repeated across the input document set. Of course, the hypothesis might be false; for instance, perhaps opinion-expressing text exhibits less variety than has previously been conjectured[5] or there is general consensus on the quality of the item under discussion.[6] But even if reviews tend to be more alike than non-reviews, the strategy outlined above would still be workable — we would only need to flip the objective/subjective labels. So the question is really whether reviews and non-reviews can be distinguished by some sort of similarity-based criterion.

In this chapter, we present three different types of "blank slate" methods for instantiating the intuition outlined above. The first approach, which clusters the documents to be categorized, and so incorporates the similarity of specific pairs of documents, can be considered as a simple baseline. The second checks within each document for material that appears in other documents. The third combines these two approaches. We find that utilizing term-distribution data across the set of documents to be classified yields quite substantial improvements over our baselines and very high accuracy overall, and that this information can sometimes prove useful in rather surprising ways.

[4] We also employ a BNC-derived term-frequency list.

[5] David and Pinch (2005) study the phenomenon of review plagiarism on sites such as Amazon.

[6] It is also conceivable that we might encounter the *Rashômon* scenario in which different authors have their own versions of the "facts", but further discussion of such a phenomenon lies beyond the scope of this work.

6.2 Related work

There have been a number of supervised and unsupervised approaches to document-level subjectivity classification proposed in previous work (Dave, Lawrence, and Pennock, 2003; Yu and Hatzivassiloglou, 2003; Wiebe et al., 2004; Finn and Kushmerick, 2006). To our knowledge, there has not been prior sentiment analysis research on using the documents to be classified as the only information source available.

The subjectivity classification problem overlaps somewhat with the area of genre classification (Karlgren and Cutting, 1994; Kessler, Nunberg, and Schütze, 1997; Argamon-Engelson, Koppel, and Avneri, 1998a); for instance, sometimes one of the genre classes corresponds to editorials (which are subjective). There is also a connection to detecting inflammatory messages, or "flames" (Spertus, 1997).

Turney (2002) proposed a knowledge-lean unsupervised approach to a sentiment-analysis problem that is related to subjectivity classification, namely, that of determining whether a subjective document conveys a positive or a negative opinion. The idea was to find words that are generally strongly associated with positive or negative sentiment by checking whether they tended to occur on the Web near the words "poor" or "excellent". Agarwal and Bhattacharyya (2005) give a knowledge-rich transductive algorithm for the same problem.

Unsupervised subjectivity classification of within-document items, such as terms, has been an active line of research; Wiebe et al. (2004) provides a survey. At the sentence level, transductive learning has also been employed (Pang and Lee, 2004).

In the area of ad hoc information retrieval, several methods have been proposed that seek to exploit similarities within the set of retrieved documents (Hearst and Pedersen, 1996; Liu and Croft, 2004; Kurland and Lee, 2005; Diaz, 2005). When

the goal is to improve the rankings of the (unknown) relevant documents within the retrieved set, taking into account each document's similarity to the (topic-based) query is very important to achieving good results (Liu and Croft, 2004; Kurland and Lee, 2005; Diaz, 2005), but it is not clear that doing so makes sense in the subjectivity-classification setting, where the query probably cannot be considered to be itself subjective.

The idea of identifying commonalities among a group of documents has been considered in the past by work on information fusion in the context of multi-document text summarization (Barzilay, McKeown, and Elhadad, 1999). The goal is to identify and eliminate redundancies in the multiple information sources, and thus enable the generation of more fluent summaries. In contrast, our interest is in utilizing cross-document redundancy as a way to characterize individual documents.

6.3 Data

We constructed a dataset intended to represent the core of the problem we are interested in, namely, separating texts on a relatively focused topic into those that are subjective and those that are objective. We chose not to work directly with search engine results in order to avoid a number of problems, both methodological (e.g., how should we develop a set of representative queries?) and pragmatic (e.g., how can we pre-process webpages to discard extraneous portions such as banner ads, and who is going to assign subjectivity labels if a large amount of data is collected?). Rather, we first started with the "Pool of 27886 unprocessed html files" dataset[7] released with the movie review data (described in Section 3.1.1), since the

[7]Available at www.cs.cornell.edu/people/pabo/movie-review-data/polarity_html.zip .

constituent documents (a) are known to be subjective, being movie reviews, and so do not require manual annotation, (b) overlap, in that there can be several reviews for the same movie, and (c) have been used by a number of groups for sentiment analysis research. From this corpus, we selected all movies for which (i) at least five reviews could be found in the Pool dataset, and (ii) plot summaries could be extracted from the Internet Movie Database and/or Yahoo! Movies. Thus, each movie represents a search topic, and the reviews and plot summaries for that movie — which we collectively term a *search set* — represent subjective and objective documents on the corresponding search topic, respectively.

Our corpus consists of a development set containing 100 search sets and a test set containing 1353 search sets. Table 6.1 gives some descriptive statistics. Note that the predominance of subjective documents reflects the fact that we are simulating the results of a query like "Novotel Sydney review".

Table 6.1: Corpus statistics, on a per-search-set basis.

	avg	min	max	median
# of subjective docs	13.5	5	83	10
# of objective docs	2.6	1	7	2

While ideally we would use a dataset with wider topic coverage, we believe our corpus still serves as a reasonable simulation of the "true" search-engine setting. At any rate, we are not aware of an alternative non-labor-intensive method for constructing a broader-coverage annotated subjectivity-classification corpus of comparable size (23,315 documents). Section 6.4.3 discusses other characteristics of our dataset that could impact the methods proposed in the next section.

6.4 Algorithms

Recall our intuition, at this point still unvalidated, that a document in a search set is likely to be objective if its contents are repeated across the search set. This section sketches two algorithms that naturally arise as attempts to exploit this conjecture, and that could still be effective even if it turns out the subjective documents resemble each other more than objective documents do.

6.4.1 Stopword removal

Before proceeding, it is important to note that common non-content-bearing terms such as "the" do not represent the same thing as objective information, but could potentially be treated the same as objective information if we simply look at frequency of occurrence. Therefore, *stopword removal* is indicated as an important pre-processing step. Unfortunately, the commonly-used InQuery stopword list (Allan et al., 2000) contains terms like "less" that, while uninformative for topic-based information retrieval, may be important for subjectivity detection. Therefore, we used a 102-item list[8] based solely on frequencies in the British National Corpus.

6.4.2 Clustering

Clustering seems to be the obvious unsupervised, blank-slate approach to distinguishing classes when at least one class is believed to contain documents that are more similar to documents in that class than to documents in the other class. Here, we group the documents in a search set into two clusters, and then, given that the data is (intentionally) skewed toward subjective documents as discussed above, we label the smaller cluster as objective. We experimented with two commonly-

[8] file http://www.eecs.umich.edu/~qstout/586/bncfreq.html.

used clustering approaches. *Single-link* hierarchical clustering can create "stringy" clusters, whereas *centroid clustering*, which successively merges the clusters whose centroids are most similar, produces clusters that are more compact. Similarity was measured via the cosine using $1 + log(\text{tf}_i)$ term weights, where tf_i is the frequency of the i^{th} term (inverse document frequency was not considered in order to clarify comparison against the method outlined in the next subsection).

It is important to point out that we do envision situations in which clustering might be ineffective. For example, the subjective documents might diverge so much from each other that they cannot be forced into one cluster.[9] We therefore considered the following alternative instantiation of the intuition outlined in the beginning of this section.

6.4.3 Idiosyncrasy-based approaches

Recall that the documents within a search set are on the same focused topic, since they are simulating responses to a specific query. As mentioned in the introduction, this setting raises the possibility that information repeated across most of the documents in a search set is likely to be factual, since opinions probably exhibit greater variability. This idea bears strong resemblance to the common information-retrieval practice of measuring inverse document frequency (idf) with respect to the entire corpus in order to downweight common terms (Robertson, 2004; Sparck Jones, 2004); but note that in our setting a factual item like a hotel's address will probably have low *corpus* idf but high *search-set* idf.

Specifically, suppose we have defined a *search-set rarity function* $\text{Rarity}_s(t)$ that varies inversely with the number of documents in the search set that contain the

[9] An alternative would be to create three clusters, which might correspond to objective documents, positive reviews, and negative reviews. However, in that case, how to make the subjective/objective assignments is not clear.

term t. Then, we define the *idiosyncrasy* score of a document x as the average search-set rarity of the terms it contains:

$$I(x) = \frac{1}{|\text{vocab}(x)|} \sum_{t \in x} \text{Rarity}_s(t), \tag{6.1}$$

where $|\text{vocab}(x)|$ is the number of different terms (types) that occur in x. If our intuitions are correct, then ranking by decreasing idiosyncrasy should put subjective documents first and objective documents last.

Defining search-set rarity

There are various ways to define a search-set rarity function. The fundamental quantity to consider is $n_s(t)$, the number of documents in the search set under consideration that contain the term t. The possibilities we explored in this work are: $\text{Rarity}_s(t) = 1/n_s(t)$, $\log(1/n_s(t))$, $1/n_s(t)^2$, or $\sqrt{1/n_s(t)}$; we refer to idiosyncrasy scoring functions based on these search-set rarity functions as *idiosyncrasy, idiosyncrasy (log), idiosyncrasy (square)*, and *idiosyncrasy (sqrt)*, respectively.

The effect of document length

One potential issue with using Equation 6.1 is that there is the possibility that factors other than what we discussed can come into play. In particular, as it happens, in our dataset the reviews are on average much longer than the plot summaries, so that classification based on length alone can actually be extremely effective; however, this is not a characteristic that we can necessarily expect to appear in other domains.[10]

Equation 6.1 normalizes the idiosyncrasy by a document's vocabulary size, which should compensate for document length. However, to further reduce the

[10] On the other hand, we can expect that *some* domains *will* share this characteristic with our dataset. For example, review-aggregation sites often place several reviews or review summaries on a single page.

influence that length might have with respect to our particular dataset, we formulated the following modified score function.[11] Let lowest-rarity-vocab$_k(x)$ be the k terms in x with the *lowest* search-set rarity (after stopword removal). Then, the idiosyncracy$_k$ score of x is

$$I_k(x) = \frac{1}{k} \sum_{t \in \text{lowest-rarity-vocab}_k(x)} \text{Rarity}_s(t). \tag{6.2}$$

We are selecting the *least* rare terms because long documents seem more likely to have terms with very high idiosyncrasy scores, and so could be selectively advantaged if it were the most rare terms that contributed to the ranking — again, our goal here is to develop a function that is as length-neutral as possible.

6.5 Evaluation

6.5.1 Preliminaries

Subjective documents form the "true" target class in the problem we are considering. However, evaluating with respect to the substantially smaller minority (objective) class provides greater dynamic range for our empirical results, and so these numbers are what we report.

An important evaluation issue is that the clustering methods we use assign objective/subjective labels to documents, whereas the idiosyncrasy-based methods give each document a score. To compare the two methods, we either have to convert the clusterings into rankings, or delineate a decision policy by which to

[11] We also looked at other ways to prevent length information from influencing the results, but all seemed unsatisfactory. Concatenating objective documents together to lengthen them changes the distribution of terms across documents within the search set quite significantly — and this distribution is, of course, the central quantity in an idiosyncrasy-based approach. Randomly removing words or sentences from reviews to shorten them also alters the term distributions significantly, and moreover could accidentally transform a review into an objective document. Shortening the reviews by applying a sentence-level subjectivity classifier to discard the most objective sentences (Pang and Lee, 2004) can preserve the subjectivity of the resultant extract, but still produces very different term distributions.

derive binary labels from the idiosyncrasy scores. We chose the former option, given that our motivating scenario is that of a search engine presenting results to a user, and search engines generally rank the documents they output. Specific conversion policies are discussed below.

The development set (100 search sets) was used only for sanity-checking the idiosyncrasy-based methods (including the value of the k parameter for idiosyncracy$_k$). Rankings are evaluated by computing the precision of the top m documents, where m is the number of objective documents in the search set. The results reported are the average over the 1353 search sets in our test set.

6.5.2 Baselines

We first consider the performance of some very simple baseline methods in order to establish a reference point for our later comparisons.

Random-choice baseline: The precision that is expected to result if we simply randomly ordered the documents is the percentage of objective documents within the search set. The average objective-document percentage over the search sets is 19.1%.

POS baseline: Past research has found that the presence of adjectives within a sentence is a strong indicator of that sentence being subjective (Hatzivassiloglou and Wiebe, 2000; Wiebe et al., 2004). This suggests a simple *non-*"blank slate" baseline (since some prior knowledge regarding subjectivity is utilized) of scoring a document according to either (a) the percentage of subjective sentences within it, assuming that a sentence containing adjectives is subjective (*adj-sentence-percent*), or (b) the percentage of adjectives within the set of tokens the document contains (*adj-percent*). The resulting precision using the Brill POS tagger (Brill, 1995)

Table 6.2: Average test-set results for objective-document precision at the number of objective documents.

random-choice baseline	19.1%
POS baseline (adj-sentence-percent)	25.1%
POS baseline (adj-percent)	39.4%
single-link (random ranking)	56.7%
centroid (random ranking)	53.5%
idiosyncrasy	71.4%
idiosyncrasy (square)	68.5%
idiosyncrasy (sqrt)	74.0%
idiosyncrasy (log)	77.5%
idiosyncracy$_k$ (flipped)	93.2%
single-link (idiosyncrasy (log) ranking)	81.7%
single-link (idiosyncracy$_k$ ranking)	86.0%
centroid (idiosyncrasy (log) ranking)	83.7%
centroid (idiosyncracy$_k$ ranking)	89.2%

was 25.1% and 39.4%, respectively. Although there is a substantial performance difference between the two, neither seems to be achieving high precision for the minority class.

6.5.3 Comparison of individual algorithms

Clustering Recall that we need to convert clusterings into rankings. Clearly, we should prefer all the documents in the smaller (presumably objective) cluster

Table 6.3: (Objective) precision and recall of the presumably objective cluster.

	precision	recall
single-link	74.4%	51.5%
centroid	85.1%	45.9%

over all the documents in the larger cluster, so the question is simply how to create within-cluster document orders. This is a non-trivial issue because of the knowledge-poor nature of the setting we are considering.

To begin with, we consider a default policy in which documents within a cluster are presented in random order. In this case, neither clustering method performs particularly well. If we look at both the objective-document precision and recall of the two methods with respect to the cluster deemed to be objective (as opposed to precision at the number of objective documents, which is our evaluation measure), we find that precision is generally higher, as shown in Table 6.3.

This indicates that a better method for within-cluster document ranking — particularly for the subjective (larger) cluster, since the recall numbers show that there are a large number of objective documents scattered through it — should lead to performance improvements. We will discuss this possibility later on.

Idiosyncrasy Of the four idiosyncrasy methods that are based on the search-set rarity of *all* the terms within a document to be scored, the log version performs the best, and the squared version performs the worst. In fact, Table 6.2 seems to show that concave-down functions of $1/n_s(t)$ are preferable, and indeed, in information retrieval, typically a log function is used to compress the differences in measured inverse document frequency for extremely rare terms.

We now turn to the idiosyncracy$_k$ scoring function (Equation 6.2), which, recall, looks at the search-set rarity only of the k least idiosyncratic terms in the document. (We used $k = 20$, which worked well for the development set.) Interestingly, this function, which we initially proposed simply as a corpus-dependent "sanity check" that reduces a potential bias towards long documents (see Section 6.4.3), gives the best overall performance by a wide margin ... *if* we choose as most objective the documents that have the *highest* idiosyncracy$_k$.

Discussion The above result, at first blush, runs completely counter to our intuitions that it should be the *subjective* documents that exhibit the most variation and thus tend to contain less frequent terms. But upon reflection, there is a reasonable basis for the phenomenon. While people's opinions may (and do) differ, what they may choose to discuss as the *basis* for their opinion may exhibit a high degree of commonality; that is, there can be common agreement on what is worth talking about, even if there is disagreement on what to think about it. As a concrete example, take the movie *The Unbearable Lightness of Being*. The most common words in the corresponding search set that appear in most of the reviews include "surgeon", "wife", and "invasion" (which are all presumably non-subjective[12]). For each of these words, there is only one objective document (out of three) that contains it. (Words that are common to both the subjective and objective classes include "Czechoslovakia" and "Sabina".)

One could make the argument that what we are seeing is still an indirect effect of the length bias in our corpus; the reasoning would be that the underlying cause for reviews being longer is that authors of reviews tend to mention (the

[12]It is possible that the term "surgeon" is being included in the reviews because it is *suggestive* of the personality type of a character in the film, although to say that it is therefore *evaluative* with respect to that character seems to be a bit of a stretch. A similar point could be made about the term "invasion".

same) auxiliary details. But describing features of the topic under discussion is a natural part of expressing a (useful) opinion, so it seems unfair to mandate that a subjectivity-classification algorithm is not allowed to make use of these descriptions. Also, as stated before, the idiosyncracy$_k$ algorithm was developed to reduce the direct effects of length in comparison to the idiosyncrasy algorithms, and the idiosyncracy$_k$ algorithm produced better results.

In any event, while different subjectivity-classification data might not exhibit the same sort of characteristics as our corpus, we believe that looking at the distribution of terms within a stopword-filtered search set should still be a practical and effective technique for separating subjective documents from objective ones.

6.5.4 Hybrid algorithms

As mentioned above, the precision of the clustering algorithms with random cluster-internal ranking, which definitively underperform the idiosyncrasy-based algorithms, ought to be improvable via the use of a better algorithm for ranking the documents within each cluster. Fortunately, we have now learned that idiosyncrasy is a good way to rank documents for subjectivity classification. This suggests a family of algorithms that combine the two approaches we have outlined above: first, apply one of the clustering techniques; then, order the documents within the resultant clusters using one of our proposed functions incorporating term search-set rarity; finally, place the "objective" (smaller) cluster above the "subjective" (larger) one.

The bottom "box" of Table 6.2 shows the results when the two best idiosyncrasy-based scoring functions are integrated into the clustering methods. Interestingly, when we use the logarithmic form for term search-set rarity, the combined algo-

rithms achieve clearly better performance (81.7% for single-link clustering, 83.7% for centroid) than any of their single components (counting clustering with random ranking as a component). In the case of idiosyncracy$_k$, the hybrid algorithm's performance represents a very large improvement over clustering alone, although it slightly underperforms using idiosyncracy$_k$ in isolation. A final observation is that for the hybrid methods, centroid clustering seems to be a better basis than single-link clustering, even though the performance ordering is reversed when these clustering algorithms are used in isolation. This can be partially explained by its higher precision with respect to the "objective" cluster (see Table 6.3).

6.6 Conclusions and Future work

We have considered three "blank-slate" methods for document-level subjectivity ranking of search sets, which models a scenario in which a search engine is being used to find reviews on a specific item of interest. Our methods were motivated by the intuition that objective documents should contain material that is frequently repeated across the search set, since such information should correspond to facts. It turned out, surprisingly, that in certain respects, subjective documents seem *more* similar to each other than objective documents are (at least for the corpus we experimented with). Yet, still, considering the distribution of terms across documents in the search set leads to very good performance (precisions ranging from the low 80's to the 90's) in this knowledge-impoverished setting.

Furthermore, we observed that after stopword removal, words with low search-set rarity may correspond to features of the matter at hand that people found worth describing or commenting on. This may lead to algorithms to automatically identify features of products in an unsupervised and knowledge-lean manner in the

context of review mining, or may be exploited to discover "hot" features that are under heavy debate (or are simply being universally noted) for a given product.

We plan to develope a dataset that covers a range of representative queries with webpages returned by actual search engines so that we can test our approaches in a more realistic setting.

Chapter 7

Unanswered questions

> *The work presents a simple yet compelling musical allegory. A string orchestra softly intones spacious chords representing the unfathomable mystery of the universe. Against this background a trumpet poses the eternal question: "Why do we exist?" Four flutes attempt to respond but cannot agree among themselves. Their growing agitation finally becomes intolerable, and they turn on the trumpet in a raucously mocking or berating fashion. In the end, the question remains unanswered, and we are left only with the harmonies of the strings, impassive and inscrutable as before.*
>
> – Paul Schiavo, program notes on Charles Ives's *The Unanswered Question*

We have presented several sentiment analysis problems and discussed the new challenges and new opportunities arising in this area. In particular, we discussed several formulations that incorporate relationship information at different levels of text corpus. The advantages of such formulations are not specific to sentiment analysis. We have already seen that it is also effective in computer vision, com-

putational biology, and other fields. In natural language processing, Many categorization problems would benefit from a means for easily incorporating contextual constraints, domain and linguistic knowledge, and class-membership probabilities derived from corpus data. We are therefore optimistic that the techniques introduced here will prove beneficial in other areas of NLP as well.

Many questions remain unanswered. For instance, can we detect sarcasm? In fact, this is one of the common questions we tend to receive: since there seems to be a high correlation between the presence of sarcasm and negative sentiment, can we use sarcasm detection to help polarity classification? The problem is it is not clear that sarcasm detection is an easier task than polarity classification when our only input is textual information. The following passage may help illustrate why:

> *"Am I* busy*?" exclaimed Authur. "Well, I've just got all these bulldozers and things to lie in front of because they'll knock my house down if I don't, but other than that ... well, no, not especially, why?"*
>
> *They don't have sarcasm on Betelgeuse, and Ford Prefect often failed to notice it unless he was concentrating. He said, "Good, is there anywhere we can talk?"*
>
> — Douglas Adams, Hitchhiker's Guide to the Galaxy

Note that even if they do have sarcasm on Betelgeuse, Ford Prefect also needs certain world knowledge such as the consequences of having one's house knocked down in order to detect the presence of sarcasm here[1].

But even before we are capable of fully modeling such world knowledge computationally, there are many interesting questions the answers to which may help

[1]Since the work described in this thesis addresses *subjective* language, it is perhaps acceptable to add a personal, speculative remark that otherwise may have been inappropriate for an academic thesis: I think we are still far from the point where it becomes critical to correctly model world knowledge.

us march towards the final steps. One that seems particularly interesting to me is whether we can effectively model the transitions in sentiment (e.g., the "thwarted expectations" narrative as discussed in Section 3.2.4) so that we can, for instance, derive a more accurate assessment of the overall sentiment expressed. A sequential document representation (such as presented in Lebanon (2006)) may be a good candidate in this case as a step towards better modeling of the discourse structure. With more sophisticated demands for information, models that go beyond the popular bag of words can become increasingly important; and as our examples have shown, sentiment analysis problems can be particularly good testing grounds for advances of this nature.

BIBLIOGRAPHY

Agarwal, Alekh and Pushpak Bhattacharyya. 2005. Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. In *Proceedings of the International Conference on Natural Language Processing (ICON)*.

Agrawal, Rakesh, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *Proceedings of WWW*, pages 529–535.

Ahuja, Ravindra, Thomas L. Magnanti, and James B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

Allan, James, Margaret E. Connell, W. Bruce Croft, Fang-Fang Feng, David Fisher, and Xiaoyan Li. 2000. INQUERY and TREC-9. In *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*, pages 551–562. NIST Special Publication 500-249.

Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of HLT/EMNLP 2005*.

Andreevskaia, Alina and Sabine Bergler. 2006. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings EACL-06, the 11rd Conference of the European Chapter of the Association for Computational Linguistics*.

Argamon, Shlomo, editor. 2003. *Proceedings of the IJCAI 2003 Workshop on DOING IT WITH STYLE: Computational Approaches to Style Analysis and Synthesis*.

Argamon, Shlomo, Jussi Karlgren, and James G. Shanahan, editors. 2005. *Proceedings of the SIGIR 2005 Workshop on Stylistic Analysis Of Text For Information Access*.

Argamon, Shlomo, Jussi Karlgren, and Ozlem Uzuner, editors. 2006. *Proceedings of the SIGIR 2006 Workshop on Stylistics for Text Retrieval in Practice*.

Argamon-Engelson, Shlomo, Moshe Koppel, and Galit Avneri. 1998a. Routing documents according to style. In *Proceedings of the International Workshop on Innovative Internet Informat ion Systems (IIIS)*, Pisa, Italy, June.

Argamon-Engelson, Shlomo, Moshe Koppel, and Galit Avneri. 1998b. Style-based text categorization: What newspaper am I reading? In *Proceedings AAAI Workshop on Text Categorization*, pages 1–4.

Atkeson, Christopher G., Andrew W. Moore, and Stefan Schaal. 1997. Locally weighted learning. *Artificial Intelligence Review*, 11(1):11–73.

Aue, Anthony and Michael Gamon. 2005a. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *Proceedings of the ACL-05 Workshop on Feature Engineering for Machine Learning in Natural Language Processing*.

Aue, Anthony and Michael Gamon. 2005b. Customizing sentiment classifiers to new domains: A case study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*.

Bansal, Nikhil, Avrim Blum, and Shuchi Chawla. 2002. Correlation clustering. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 238–247. Journal version in *Machine Learning Journal*, special issue on theoretical advances in data clustering, 56(1-3):89–113 (2004).

Barzilay, Regina and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In *Proceedings of HLT/EMNLP*, pages 331–338.

Barzilay, Regina, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *37th Annual Meeting of the Association for Computational Linguistics*.

Beineke, Philip, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. 2004. Exploring sentiment summarization. In Qu et al. (Qu, Shanahan, and Wiebe, 2004). AAAI technical report SS-04-07.

Beineke, Philip, Trevor Hastie, and Shivakumar Vaithyanathan. 2004. The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the ACL*, pages 263–270, July.

Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Biber, Douglas. 1988. *Variation across Speech and Writing*. Cambridge University Press.

Blum, Avrim and Shuchi Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of ICML*, pages 19–26.

Boykov, Yuri, Olga Veksler, and Ramin Zabih. 1999. Fast approximate energy minimization via graph cuts. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 377–384. Journal version in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 23(11):1222–1239, 2001.

Breck, Eric and Claire Cardie. 2004. Playing the telephone game: Determining the hierarchical structure of perspective and speech expressions. In *Proceedings of the 20th COLING*.

Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Cardie, Claire, Cynthia Farina, Thomas Bruce, and Erica Wagner. 2006. Using natural language processing to improve eRulemaking. In *Proceedings of Digital Government Research (dg.o)*.

Cardie, Claire, Janyce Wiebe, Theresa Wilson, and Diane Litman. 2003. Combining low-level and summary representations of opinions for multi-perspective question answering. In *AAAI Spring Symposium on New Directions in Question Answering*, pages 20–27.

Carvalho, Vitor and William W. Cohen. 2005. On the collective classification of email "speech acts". In *Proceedings of SIGIR*, pages 345–352.

Chen, Stanley and Ronald Rosenfeld. 2000. A survey of smoothing techniques for ME models. *IEEE Trans. Speech and Audio Processing*, 8(1):37–50.

Choi, Yejin, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of HLT-EMNLP-05, the Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing*.

Collins-Thompson, Kevyn and Jamie Callan. 2004. A language modeling approach to predicting reading difficulty. In *HLT-NAACL: Proceedings of the Main Conference*, pages 193–200.

Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 1990. *Introduction to Algorithms*. MIT Press.

Daelemans, Walter and Véronique Hoste. 2002. Evaluation of machine learning methods for natural language processing tasks. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 755–760.

Das, Sanjiv and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*.

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

David, Shay and Trevor John Pinch. 2005. Six degrees of reputation: The use and abuse of online review and recommendation systems. Available at the Social Science Research Network, http://ssrn.com/abstract=857505.

Della Pietra, Stephen, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

Dhillon, Inderjit. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD Conference*, pages 269–274.

Diaz, Fernando. 2005. Regularizing ad hoc retrieval scores. In *Proceedings of the Fourteenth International Conference on Information and Knowledge Managment (CIKM)*, pages 672–679.

Domingos, Pedro and Michael J. Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

Edoardo M. Airoldi, Xue Bai, Rema Padman. 2006. Markov blankets and meta-heuristic search: Sentiment extraction from unstructured text. *Lecture Notes in Computer Science*, 3932.

Efron, Miles. 2004. Cultural orientation: Classifying subjective documents by cociation [sic] analysis. In *Proceedings of the AAAI Fall Symposium on Style and Meaning in Language, Art, Music, and Design*, pages 41–48.

Ekman, Paul. 1982. *Emotion in the Human Face. Second edition.* Cambridge University Press.

Engström, Charlotta. 2004. Topic dependence in sentiment classification. Master's thesis, University of Cambridge.

Esuli, Andrea. 2006. Sentiment classification bibliography. liin-www.ira.uka.de/bibliography/Misc/Sentiment.html.

Finn, Aidan and Nicholas Kushmerick. 2006. Learning to classify documents according to genre. *Journal of the American Society for Information Science and Technology (JASIST)*, 7(5). Special issue on computational analysis of style.

Finn, Aidan, Nicholas Kushmerick, and Barry Smyth. 2002. Genre classification and domain transfer for information filtering. In *Proceedings European Colloquium on Information Retrieval Research*, pages 353–362, Glasgow.

Galley, Michel, 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 ACL*, pages 669–676.

Gamon, Michael. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceeding of COLING-04, the 20th International Conference on Computational Linguistics*.

Gaussier, Éric. 1998. Flow network models for word alignment and terminology extraction from bilingual corpora. In *Proceedings of COLING/ACL*, pages 444–450.

Getoor, Lise, Nir Friedman, Daphne Koller, and Benjamin Taskar. 2002. Learning probabilistic models of relational structure. *Journal of Machine Learning Research*, 3:679–707. Special issue on the Eighteenth ICML.

Goldberg, Andrew B. and Jerry Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *TextGraphs: HLT/NAACL Workshop on Graph-based Algorithms for Natural Language Processing*.

Grefenstette, Gregory, Yan Qu, James G. Shanahan, and David A. Evans. 2004. Coupling niche browsers and affect analysis for an opinion mining application. In *Proceedings of RIAO*.

Hatzivassiloglou, Vasileios and Kathleen McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th ACL/8th EACL*, pages 174–181.

Hatzivassiloglou, Vasileios and Janyce Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of COLING*.

Hearst, Marti. 1992. Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum Associates, pages 257–274.

Hearst, Marti A. and Jan O. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR*.

Herbrich, Ralf, Thore Graepel, and Klaus Obermayer. 2000. Large margin rank boundaries for ordinal regression. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, Neural Information Processing Systems. MIT Press, pages 115–132.

Hillard, Dustin, Mari Ostendorf, and Elizabeth Shriberg. 2003. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proceedings of HLT-NAACL*.

Horvitz, Eric, Andy Jacobs, and David Hovel. 1999. Attention-sensitive alerting. In *Proceedings of the Conference on Uncertainty and Artificial Intelligence*, pages 305–313.

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

Huettner, Alison and Pero Subasic. 2000. Fuzzy typing for document management. In *ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes*, pages 26–27.

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

Ishikawa, Hiroshi. 2003. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10).

Ishikawa, Hiroshi and Davi Geiger. 1998. Occlusions, discontinuities, and epipolar lines in stereo. In *Proceedings of the 5th European Conference on Computer Vision (ECCV)*, volume I, pages 232–248, London, UK. Springer-Verlag.

Joachims, Thorsten. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 137–142.

Joachims, Thorsten. 1999. Making large-scale SVM learning practical. In Bernhard Schölkopf and Alexander Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, pages 44–56.

Joachims, Thorsten. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of ICML*, pages 290–297.

Kamps, Jaap, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. 2004. 2004. Using wordnet to measure semantic orientation of adjectives. In *LREC*.

Karlgren, Jussi and Douglass Cutting. 1994. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of COLING*, pages 1071–1075.

Kessler, Brett, Geoffrey Nunberg, and Hinrich Schütze. 1997. Automatic detection of text genre. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 32–38.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING*.

Kim, Soo-Min and Eduard Hovy. 2005a. Automatic detection of opinion bearing words and sentences. In *Companion Volume to the Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-05)*.

Kim, Soo-Min and Eduard Hovy. 2005b. Identifying opinion holders for question answering in opinion texts. In *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*.

Kleinberg, Jon and Éva Tardos. 2002. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639.

Kobayashi, Nozomi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. 2004. Collecting evaluative expressions for opinion extraction. In *Proceedings of IJCNLP-04, the 1st International Joint Conference on Natural Language Processing*.

Kondor, Risi Imre and John D. Lafferty. 2002. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of ICML*, pages 315–322.

Koppel, Moshe and Jonathan Schler. 2005. The importance of neutral examples for learning sentiment. In *Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations (FINEXIN)*.

Kurland, Oren and Lillian Lee. 2005. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR*, pages 306–313.

Kwon, Namhee, Stuart Shulman, and Eduard Hovy. 2006. Multidimensional text analysis for eRulemaking. In *Proceedings of Digital Government Research (dg.o)*.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Laver, Michael, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American Political Science Review*.

Lebanon, Guy. 2006. Sequential document representations and simplicial curves. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*.

Lee, Yong-Bae and Sung Hyon Myaeng. 2002. Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*.

Lehnert, Wendy, Claire Cardie, and Ellen Riloff. 1990. Analyzing research papers using citation sentences. In *Program of the Twelfth Annual Conference of the Cognitive Science Society*, pages 511–18.

Lewis, David D. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. pages 4–15. Invited talk.

Lin, Wei-Hao, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on? identifying perspectives at the document and sentence levels. In *CoNLL*.

Liu, Hugo, Henry Lieberman, and Ted Selker. 2003. A model of textual affect sensing using real-world knowledge. In *Proceedings of Intelligent User Interfaces (IUI)*, pages 125–132.

Liu, Xiaoyong and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of SIGIR*, pages 186–193.

Malioutov, Igor and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the Association for Computational Linguistics Conference (ACL-2006)*.

Marcu, Daniel. 2000. *The theory and practice of discourse parsing and summarization*. MIT Press.

Matsumoto, Shotaro, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment classification using word sub-sequences and dependency sub-trees. In *Proceedings of PAKDD'05, the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*.

McCallum, Andrew. 1999. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*.

McCallum, Andrew and Kamal Nigam. 1998. A comparison of event models for Naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48.

McCallum, Andrew and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of NIPS*.

McCullagh, Peter. 1980. Regression models for ordinal data. *Journal of the Royal Statistical Society*, 42(2):109–42.

Mihalcea, Rada and Carlo Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Journal of Computational Intelligence*.

Mosteller, Frederick and David L. Wallace. 1984. *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*. Springer-Verlag.

Mullen, Tony and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, pages 412–418, July. Poster.

Mullen, Tony and Robert Malouf. 2006. A preliminary investigation into sentiment analysis of informal political discourse. In *Proceedings of the AAAI Symposium on Computational Approaches to Analyzing Weblogs*, pages 159–162.

Munson, Art, Claire Cardie, and Rich Caruana. 2005. Optimizing to arbitrary NLP metrics using ensemble selection. In *Proceedings of HLT-EMNLP*, pages 539–546.

Neville, Jennifer and David Jensen. 2000. Iterative classification in relational data. In *Proceedings of the AAAI Workshop on Learning Statistical Models from Relational Data*, pages 13–20.

Nicolov, Nicolas, Franco Salvetti, Mark Liberman, and James H. Martin, editors. 2006. *the AAAI Spring Symposium on Computational Approaches to Weblogs*. AAAI Press.

Nigam, Kamal, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

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

Pang, Bo and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Pang, Bo and Lillian Lee. 2008. Using very simple statistics for review search: An exploration. In *Proceedings of the International Conference on Computational Linguistics (COLING)*. Poster.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Pedersen, Ted. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings Second NAACL*, pages 79–86.

Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP*.

Popescu, Ana-Maria, Oren Etzioni, and Henri Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of IUI*.

Purpura, Stephen and Dustin Hillard. 2006. Automated classification of congressional legislation. In *Proceedings of Digital Government Research (dg.o)*.

Qu, Yan, James Shanahan, and Janyce Wiebe, editors. 2004. *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. AAAI Press. AAAI technical report SS-04-07.

Rabiner, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), February.

Read, Jonathon. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*.

Rifkin, Ryan M. and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141.

Riloff, Ellen and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of EMNLP*.

Robertson, Stephen E. 2004. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

Sack, Warren. 1994. On the computation of point of view. In *Proceedings of AAAI*, page 1488. Student abstract.

Schapire, Robert E. and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Shulman, Stuart, Jamie Callan, Eduard Hovy, and Stephen Zavestoski. 2005. Language processing technologies for electronic rulemaking: A project highlight. In *Proceedings of Digital Government Research (dg.o)*, pages 87–88.

Shulman, Stuart and David Schlosberg. 2002. Electronic rulemaking: New frontiers in public participation. Prepared for the Annual Meeting of the American Political Science Association.

Smith, Steven S., Jason M. Roberts, and Ryan J. Vander Wielen. 2005. *The American Congress*. Cambridge University Press, fourth edition.

Smola, Alex J. and Bernhard Schölkopf. 1998. A tutorial on support vector regression. Technical Report NeuroCOLT NC-TR-98-030, Royal Holloway College, University of London.

Sparck Jones, Karen. 2004. IDF term weighting and IR research lessons. *Journal of Documentation*, 60(5):521–523.

Spertus, Ellen. 1997. Smokey: Automatic recognition of hostile messages. In *Proc. of Innovative Applications of Artificial Intelligence (IAAI)*, pages 1058–1065.

Stamatatos, E., N. Fakotakis, and G. Kokkinakis. 2000. Text genre detection using common word frequencies. In *Proceedings of the 18th International Conference on computational Linguistics (ACL)*.

Stolcke, Andreas, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Stoyanov, Veselin, Claire Cardie, Diane Litman, and Janyce Wiebe. 2004. Evaluating an opinion annotation using a new multi-perspective question and answer corpus. In James G. Shanahan, Janyce Wiebe, and Yan Qu, editors, *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Stanford, US.

Subasic, Pero and Alison Huettner. 2001. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems*, 9(4):483–496.

Takamura, Hiroya, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Takamura, Hiroya, Yuji Matsumoto, and Hiroyasu Yamada. 2004. Modeling category structures with a kernel function. In *Proceedings of CoNLL*, pages 57–64.

Taskar, Ben, Pieter Abbeel, and Daphne Koller. 2002. Discriminative probabilistic models for relational data. In *Proceedings of UAI*, Edmonton, Canada.

Taskar, Ben, Vassil Chatalbashev, and Daphne Koller. 2004. Learning associative Markov networks. In *Proceedings of ICML*.

Taskar, Ben, Carlos Guestrin, and Daphne Koller. 2003. Max-margin Markov networks. In *Proceedings of NIPS*.

Terveen, Loren, Will Hill, Brian Amento, David McDonald, and Josh Creter. 1997. PHOAKS: A system for sharing recommendations. *Communications of the ACM*, 40(3):59–62.

Teufel, Simone and Marc Moens. 2002. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Thomas, Matt, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of EMNLP*.

Tomokiyo, Laura Mayfield and Rosie Jones. 2001. You're not from round here, are you? Naive Bayes detection of non-native utterance text. In *Proceedings Second NAACL*, pages 239–246.

Tong, Richard M. 2001. An operational system for detecting and tracking opinions in on-line discussion. SIGIR Workshop on Operational Text Classification.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the ACL*, pages 417–424.

Turney, Peter D. and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

Vapnik, Vladimir. 1995. *The Nature of Statistical Learning Theory*. Springer.

Wiebe, Jan and Rada Mihalcea. 2006. Word sense and subjectivity. In *Proceedings of the Conference on Computational Linguistics / Association for Computational Linguistics (COLING/ACL)*.

Wiebe, Janyce and Theresa Wilson. 2002. Learning to disambiguate potentially subjective expressions. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 112–118.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2):0–0.

Wiebe, Janyce M. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Wiebe, Janyce M. and William J. Rapaport. 1988. A computational theory of perspective and reference in narrative. In *Proceedings of the ACL*, pages 131–138.

Wiebe, Janyce M., Theresa Wilson, and Matthew Bell. 2001. Identifying collocations for recognizing opinions. In *Proceedings of the ACL/EACL Workshop on Collocation*.

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

Wilks, Yorick and Mark Stevenson. 1998. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Journal of Natural Language Engineering*, 4(2):135–144.

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

Wilson, Theresa, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of AAAI*, pages 761–769.

Yang, Hui and Jamie Callan. 2005. Near-duplicate detection for eRulemaking. In *Proceedings of Digital Government Research (dg.o)*.

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

Yi, Jeonghee and Wayne Niblack. 2005. Sentiment mining in webfountain. In *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP*.

Zhu, Jerry. 2005a. Semi-supervised learning literature survey. Computer Sciences Technical Report TR 1530, University of Wisconsin-Madison. Available at http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf; has been updated since the initial 2005 version.

Zhu, Xiaojin (Jerry). 2005b. *Semi-Supervised Learning with Graphs*. Ph.D. thesis, Carnegie Mellon University.