Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification

Prem Melville
IBM T.J. Watson Research Ctr.
P.O. Box 218
Yorktown Heights, NY 10598
pmelvil@us.ibm.com

Wojciech Gryc
Oxford Univ. Computing Lab
Wolfson Bldg, Parks Rd
Oxford OX1 3QD, UK
wojciech.gryc@trinity.ox.ac.uk

Richard D. Lawrence
IBM T.J. Watson Research Ctr.
P.O. Box 218
Yorktown Heights, NY 10598
ricklawr@us.ibm.com

ABSTRACT

The explosion of user-generated content on the Web has led to new opportunities and significant challenges for companies, that are increasingly concerned about monitoring the discussion around their products. Tracking such discussion on weblogs, provides useful insight on how to improve products or market them more effectively. An important component of such analysis is to characterize the sentiment expressed in blogs about specific brands and products. Sentiment Analysis focuses on this task of automatically identifying whether a piece of text expresses a positive or negative opinion about the subject matter. Most previous work in this area uses prior lexical knowledge in terms of the sentiment-polarity of words. In contrast, some recent approaches treat the task as a text classification problem, where they learn to classify sentiment based only on labeled training data. In this paper, we present a unified framework in which one can use background lexical information in terms of word-class associations, and refine this information for specific domains using any available training examples. Empirical results on diverse domains show that our approach performs better than using background knowledge or training data in isolation, as well as alternative approaches to using lexical knowledge with text classification.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models

General Terms

Algorithms, Economics, Experimentation

1. INTRODUCTION

Over the past decade, the Internet has created enormous opportunities for companies of all sizes to reach customers, advertise products, and transact business. In this well established business model, companies can largely control their own web-based reputation via the content appearing on their websites. Web 2.0, with its emphasis on sites that promote information sharing and user collaboration, is quickly altering this landscape. Increasingly, the focal point of discussion of all aspects of a company's product portfolio is moving from individual company websites to collaborative sites, blogs, and forums where essentially anyone can post comments that may influence perceptions and purchase behavior of a large number of potential buyers. This is of obvious concern to marketing organizations, not only because the spread of negative information can be difficult to control, but because it can be very difficult to even detect it given the huge growth in blogs, forums, and other Web 2.0 phenomena. This concern has given rise to the term "buzz", and several sites (e.g. [2]) and systems (e.g. [27]) have been introduced to monitor trends in the blog-based reputation of specific companies and product brands.

The automated analysis of blog-related buzz raises several interesting questions from a marketing perspective:

1. Given a huge number (order 100 million) of blogs, how can we identify the subset of blogs and forums that are discussing not only a specific product, but higher-level concepts that are in some way relevant to this product?

2. Having identified this subset of relevant blogs, how do we identify the most authoritative or influential bloggers in this space?

3. How do we detect and characterize specific sentiment expressed about an entity (e.g. product) mentioned in a blog or forum?

Each of these marketing questions raises interesting technical issues for the Data Mining community, which we are pursuing as part of a broader agenda. In this paper, we focus only on the problem of sentiment analysis. In the context of blog analysis, the objective of sentiment analysis is to determine the overall attitude expressed in a portion of text contained within a blog. The text can be either an entire blog post, or a snippet extracted in the proximity of the mention of a specific entity such as a company or product.

Most prior work in sentiment analysis use knowledge-based approaches, that classify the sentiment of texts based on dictionaries defining the sentiment-polarity of words, and simple linguistic patterns. Recently, there have been some studies that take a machine learning approach [20, 8], and build text classifiers trained on documents that have been human-labeled as *positive* or *negative*. The knowledge-based

KDD'09, June 28–July 1, 2009, Paris, France.

approaches do not adapt well to different domains, while the learning approaches require much effort in human annotation of documents. In this paper, we present a new machine learning approach that overcomes these drawbacks by effectively combining background lexical knowledge with supervised learning. In particular, we construct a generative model based on a lexicon of sentiment-laden words, and a second model trained on labeled documents. The distributions from these two models are then adaptively pooled to create a composite multinomial Naïve Bayes classifier that captures both sources of information. By exploiting prior lexical knowledge we dramatically reduce the amount of training data required. In addition, by using some labeled documents we are able to refine the background knowledge, which is based on a generic lexicon, thus effectively adapting to new domains. We demonstrate the generality of our approach on three, very different domains — blogs discussing enterprise-software products, political blogs discussing US Presidential candidates, and online movie reviews. Empirical results show that our approach of Pooling Multinomials performs better than various baselines that use lexical knowledge and labeled data in isolation, and an alternative approach to using background knowledge in a semi-supervised setting.

2. BASELINE APPROACHES

For the purpose of this paper we treat sentiment detection as the binary polarity classification of documents into positive and negative sentiment classes. In this section, we describe two baseline approaches to using background knowledge in such a classification of documents.

2.1 Lexical classification

In the absence of any labeled data in a domain, one can build sentiment-classification models that rely solely on background knowledge, such as a lexicon defining the polarity of words. Given a lexicon of positive and negative terms, one straightforward approach to using this information is to measure the frequency of occurrence of these terms in each document. The probability that a test document D belongs to the positive class can then be computed as $P(+|D) = \frac{a}{a+b}$; where a and b are the number of occurrences of positive and negative terms in the document respectively. A document is then classified as positive if $P(+|D) > t$, where t is the classification threshold; otherwise, the document is classified as negative. In the absence of any prior information of the relative positivity and negativity of terms we use $t = 0.5$, i.e. we assume a document is positive if there are more positive terms than negative in the document. We refer to this classification approach as the Lexical Classifier, and use it as one of our baselines.

For this study, we used a lexicon generated by the IBM India Research Labs that was developed for other text mining applications [22]. It contains 2,968 words that have been human-labeled as expressing positive or negative sentiment. In total, there are 1,267 positive and 1,701 negative unique terms after stemming. It should be noted, that this list was constructed without a specific domain in mind; which is further motivation for using training examples to learn domain-specific connotations.

2.2 Feature supervision

An alternative to a purely-lexical classification scheme is to use a lexicon along with unlabeled data in a semi-supervised learning setting. There have been few approaches to incorporating such background knowledge into learning, which we list in Section 5. Most of these approaches create *pseudo examples* based on the background knowledge, and then use existing learning algorithms. Here, we describe the approach of Liu et al. [15] since it is the most closely related to our work; as they too use Naïve Bayes classification and make use of background knowledge via word-class associations (i.e., labeled features). However, unlike us, they use unlabeled documents in learning. Given a representative set of words for each class (i.e., a lexicon), they create a *representative document* for each class containing all the representative words. Then they compute the cosine similarity between each document in the unlabeled set with the representative documents. They assign each unlabeled document to the class with the highest similarity, and then train a Naïve Bayes classifier using these pseudo-labeled examples. Liu et al. demonstrate that their approach performs better than using only a Lexical Classifier. We refer to their approach as Feature Supervision and present it as a baseline to using labeled features along with a supervised learning approach.

3. POOLING MULTINOMIALS

Pang et al. [20] have shown that using only lexical information is not as effective for sentiment classification as building machine learning models from training examples. However, completely ignoring the background knowledge provided by a lexicon, in lieu of training data, may not be optimal. As an alternative, we propose the Pooling Multinomials classifier that provides a framework in which one can build a composite Naïve Bayes classifier that incorporates both background knowledge and training examples.

Below, we present some basic concepts on multinomial Naïve Bayes text classification, required for the description of our framework. More details on event models for text classification and induction of Naïve Bayes classifiers can be found in [16].

The multinomial Naïve Bayes classifier commonly used for text categorization relies on three assumptions [18]: (1) documents are produced by a mixture model; (2) there is a one-to-one correspondence between each mixture component and class, and (3) each mixture component is a multinomial distribution of words, i.e., given a class, the words in a document are produced independently of each other.

Based on this generative model, the likelihood of a document (D) is the sum of total probability over all mixture components, i.e., $P(D) = \sum_j P(D|c_j)P(c_j)$; where $P(c_j)$ is the probability of the class c_j, and $P(D|c_j)$ is the probability of the document given the class. We refer to the set of classes as $\mathcal{C}$, which for binary sentiment classification is $\{+, -\}$. In computing the likelihood of a document, we make the naive assumption that the words (w_i) of a document are generated independently, so the probability of a document, D, being generated in a given class, c_j, is $P(D|c_j) = \prod_i P(w_i|c_j)$

The Naïve Bayes classification rule uses Bayes' theorem to compute the class membership probabilities of each class,

$$P(c_j|D) = \frac{P(c_j)\prod_i P(w_i|c_j)}{P(D)} \quad (1)$$

and the class with the highest likelihood is predicted, i.e.,

$$argmax_{c_j} P(c_j) \prod_i P(w_i|c_j) \quad (2)$$

3.1 Combining probability distributions

Pooling distributions is a general approach for combining information from multiple sources or experts; where experts are typically represented in terms of probability distributions. The combination of such expert probability distributions has been an area of much study in the field of Risk Analysis [3]. Since we are dealing with text categorization, an "expert" in our setting can be represented as multinomial probability distributions as done in Naïve Bayes classification. We consider two experts — one learned based on labeled training data, and the other representing a generative model that explains the lexicon. We will discuss the latter, background-knowledge model in the next section; but for now assume we have the multinomial parameters of such a model.

In this paper, we only consider the case of two sources of information, i.e., labeled examples in the target domains and background lexical knowledge. However, Pooling Multinomials is a general framework that can be used to combine multiple multinomial models — these could be derived from training data from different related domains or different sources of background knowledge. There is a substantial literature on the mathematical combination of probability distributions, including reviews such as Winkler [30], Genest and Zidek [11], and French [10]. Here, we compare two axiomatic approaches, namely the *linear opinion pool* and the *logarithm opinion pool*.

The *linear opinion pool* is an appealing approach to aggregating probability distributions, which dates back to Laplace [3]. In this approach the aggregate probability:

$$P(w_i|c_j) = \sum_{k=1}^{K} \alpha_k P_k(w_i|c_j) \quad (3)$$

where K is the number of experts; $P_k(w_i|c_j)$ represents the probability assigned by expert k to word w_i occurring in a document of class c_j; and the weights α_k sum to one.

Another typical combination approach is the *logarithmic opinion pool*, which uses multiplicative averaging. In this approach the combined probability:

$$P(w_i|c_j) = Z \prod_{k=1}^{K} P_k(w_i|c_j)^{\alpha_k} \quad (4)$$

where Z is a normalizing constant, and weights α_k satisfy restrictions that assure that $P(w_i|c_j)$ is a probability distribution. Typically, the weights are restricted to sum to one. If the weights are equal $(1/K)$ then Log Pooling is equivalent to taking the geometric mean of the individual distributions. The scheme is called Log Pooling because the logarithm of the combined distribution can be expressed as a linear combination of the logarithms of the individual distributions.

For both pooling schemes, we compute weights of individual experts based on their error in explaining the training data. In particular we use a sigmoid weighting scheme, where:

$$\alpha_k = \log \frac{1 - err_k}{err_k} \quad (5)$$

where err_k is the error of expert k on the training set; and the α_k's are normalized to sum to one. This is the same weighting scheme used to combine additive models in boosting [23].

To learn a model from the training data we compute conditionals $P(w_i|c_j)$ based on observed term frequencies, as in standard Naïve Bayes classification. In addition, for Pooling Multinomials we need to construct a multinomial model representing the background knowledge, which we describe in detail in the next section.

3.2 A generative background-knowledge model

Inducing a multinomial Naïve Bayes classifier involves estimating the model parameters $P(C_j)$ and $P(w_i|c_j)$. In the absence of background knowledge about the class distribution, we estimate the class priors $P(C_j)$ from the training data. So for the background model we only focus on the conditional probabilities of each word given the class. We assume that the feature-class associations provided in the lexicon are implicitly arrived at by the human experts by examining many positive and negative sentiment documents. So we attempt to select the parameters $P(w_i|c_j)$ of the multinomial distributions that would generate such documents. The exact values of these conditionals are derived below, based on a set of properties these distributions must satisfy. To aid our derivations, we first establish some important notation below.

Definitions:
$\mathcal{V}$ – the vocabulary, i.e., set of words in our domain
$\mathcal{P}$ – set of positive terms from the lexicon that exists in $\mathcal{V}$
$\mathcal{N}$ – set of negative terms from the lexicon that exists in $\mathcal{V}$
$\mathcal{U}$ – set of unknown terms, i.e. $\mathcal{V} - (\mathcal{N} \cup \mathcal{P})$
m – size of vocabulary, i.e. $|\mathcal{V}|$
p – number of positive terms, i.e. $|\mathcal{P}|$
n – number of negative terms, i.e. $|\mathcal{N}|$

Property 1: Since we do not know the relative polarity of terms in the dictionary, we assume all positive terms are equally likely to occur in a positive document, and the same is true for negative documents, i.e.,

$$P(w_i|+) = P(w_j|+), \forall w_i, w_j \in \mathcal{P}$$
$$\text{and } P(w_i|-) = P(w_j|-), \forall w_i, w_j \in \mathcal{N} \quad (6)$$

We refer to the probability of any positive term appearing in a positive document simply as $P(w_+|+)$. Similarly, we refer to the probability of any negative term appearing in a negative document as $P(w_-|-)$.

Furthermore, in the absence of any knowledge about words that are not in our lexicon, we treat them equally in each class, i.e.,

$$P(w_i|+) = P(w_j|+), \forall w_i, w_j \in \mathcal{U}$$
$$\text{and } P(w_i|-) = P(w_j|-), \forall w_i, w_j \in \mathcal{U} \quad (7)$$

We refer to these conditional probabilities of non-lexicon terms as $P(w_u|+)$ and $P(w_u|-)$.

Property 2: If a document D_i has α positive terms and β negative terms, and document D_j has β positive terms and α negative terms, we would like D_i to be considered as likely to be a positive document, as D_j is likely to be a negative document. Using (1) and this property of symmetry gives

us the following requirement:

$$[P(w_+|+)]^\alpha [P(w_-|+)]^\beta = [P(w_-|-)]^\alpha [P(w_+|-)]^\beta$$
$$\Rightarrow \left[\frac{P(w_+|+)}{P(w_-|-)}\right]^\alpha = \left[\frac{P(w_+|-)}{P(w_-|+)}\right]^\beta$$

For this to be true for all values of α and β, we need

$$P(w_+|+) = P(w_-|-)$$
$$\text{and } P(w_-|+) = P(w_+|-) \quad (8)$$

That is, the probability of a positive term appearing in a positive document is the same as that of a negative term appearing in a negative document; and the same is true for the conditional probabilities of terms occurring in documents with opposite polarity.

Property 3: Since a positive document is more likely to contain a positive term than a negative term, and vice versa, we would like:

$$P(w_+|+) = r \times P(w_-|+)$$
$$\text{and } P(w_-|-) = r \times P(w_+|-)$$
$$\text{where } 0 < 1/r \leq 1 \quad (9)$$

We refer to the factor r as the *polarity level* — it is a measure of how much more likely it is for a positive term to occur in a positive document compared to a negative term.

Property 4: Since each component of our mixture model is a probability distribution, we have the following constraint on the conditional probabilities for each class, c_j:

$$\sum_i^m P(w_i|c_j) = 1 \quad (10)$$

Using the above properties as constraints, we can now derive the appropriate values to use for our background-knowledge model.

From (10) it follows that

$$pP(w_+|+) + nP(w_-|+) + (m-p-n)P(w_u|+) = 1 \quad (11)$$

Which gives us the following inequality

$$pP(w_+|+) + nP(w_-|+) \leq 1$$
$$\Rightarrow pP(w_+|+) + n\frac{P(w_+|+)}{r} \leq 1$$

using (9).

Since $0 < 1/r \leq 1$, it follows that,

$$P(w_+|+) \leq \frac{1}{p+n}$$

Since we want to assign the maximum probability mass to the known terms in the lexicon, we set $P(w_+|+)$ to the maximum value possible, i.e.

$$P(w_+|+) = \frac{1}{p+n} \quad (12)$$

Now, it follows from (8) and (9) that

$$P(w_-|-) = \frac{1}{p+n}$$
$$P(w_+|-) = P(w_-|+) = \frac{1}{p+n} \times \frac{1}{r} \quad (13)$$

Now, solving for (11) (and analogously for the negative class), we get the following conditionals for the unknown terms:

$$P(w_u|+) = \frac{n(1-1/r)}{(p+n)(m-p-n)}$$
$$P(w_u|-) = \frac{p(1-1/r)}{(p+n)(m-p-n)} \quad (14)$$

Using (12), (13) and (14), we now have all the requisite parameters to represent our background knowledge.

4. EMPIRICAL EVALUATION

In this section, we present experiments and analysis of the application of Pooling Multinomials to sentiment analysis in different domains.

4.1 Data sets

As discussed earlier, our motivating application is to automate the analysis of blog posts as they relate to product and/or brand names. For this purpose, we have created a set of labeled sentiment examples related to the IBM Lotus software brand. While we are primarily interested in the sentiment of technology blogs, we have also applied our methodology to characterize sentiment in blogs discussing specific political candidates.

One non-trivial part of blog data collection for sentiment analysis is the extraction of the *relevant* text from the downloaded website. Blogs are by nature much more diverse in layout and structure than movie or product reviews. In addition, many blogs have significant numbers of comments (often from individuals other than the blogger) as well as explicit citations. Both comments and citation often exhibit the exact opposite sentiment from the main content. However, the automated distinction between core content, citations, and comments is very difficult. We use the algorithm provided by [9] to extract text only from parts of the Web-page where the ratio of HTML tags to words is above a minimal threshold.

Lotus blogs: Our target application is detecting sentiment around enterprise software, specifically IBM Lotus collaborative software. In order to do this, we have been monitoring 20,488 technology blogs which currently contain over 1.7 million posts. The labeled *Lotus* data set we created, consists of posts from 14 of these blogs, 4 of which are actively posting negative content on the brand, with the rest tending to write more positive or neutral posts. In this data set, negative blog posts often complain about user interface challenges or software bugs. For example, a comment like "Could someone please tell me why Lotus notes takes 99% of my CPU usage?" could be seen as negative, while "DAMO demo provides a list of reason to go Lotus" is positive. The *Lotus* data was collected by downloading the latest posts from each blogger's RSS feeds, or accessing the blog's archives, if they exist. Each post was then read and labeled by hand as either positive, negative, neutral, or not relevant. For our analysis, only positive and negative posts were retained, creating a labeled set of 34 and 111 examples, respectively. Since some bloggers tend to exhibit consistently positive or negative sentiment, only the body of every post was used in the analysis, thus avoiding blog titles and recurring information like user names, which may ultimately lead to over-training of the models.

Political candidate blogs: Posts focusing on politics were taken from a continuously updated set of 16,741 political blogs, which contain over 2 million posts. We focused our labeling effort on randomly selected posts containing the term "Obama" or "Clinton" in their URLs. Unlike the Lotus-focused posts, the political posts all come from diverse blogs. Furthermore, from the experience of human labelers, it appears that political sentiment is much more difficult to label than software reviews, as posts tend to be more emotional, discuss issues only implicitly related to candidates (e.g. economic or foreign policies), and may also use cultural references to pass judgment. A post was labeled as having positive or negative sentiment about a specific candidate (Barack Obama or Hillary Clinton) if it explicitly mentioned the candidate in positive or negative terms. Objective statements and quotations from newspapers and other sources were ignored. Similarly, if the blogger made implicit statements about a candidate (e.g. discussing racism or sexism in elections without specifically mentioning a candidate), that post would not be associated with sentiment. Essentially, only posts with clear opinions about a candidate were labeled and included in this analysis. For example, "I think Hillary Clinton is not doing well in this debate." would be a labeled as negative for Clinton. On the other hand, "Obama names top fund-raisers, gives more details than Clinton." would be seen as neutral because the reader cannot assign sentiment without making a personal value judgment on the statement. Throughout labeling, only positive and negative posts were retained – those labeled as neutral and not relevant were discarded. Similarly, if a post was seen as negative about one candidate and positive about another, then the post was also discarded. While discarding posts in such a manner is not an option if one were to deploy a classifier in a production environment, such a rigorous process of post selection was used to build a clean test set to evaluate our methods. The final *Politics* data set consisted of 49 positive and 58 negative posts.

Movie reviews: Apart from the blog data that we generated, we also used the publicly available data of movie reviews provided by Pang et al. [20]. This data consists of 1000 positive and 1000 negative reviews from the Internet Movie Database. Positive labels were assigned to reviews that had a rating above 3.5 stars and negative labels were assigned to the rest.

4.2 Results

We compare our approaches, Linear Pooling and Log Pooling to the Lexical Classifier, Feature Supervision as in Liu et al. [15], and to using only the training data with a Naïve Bayes classifier as in Pang et al. [19]. For the Naïve Bayes model, we compute the conditional probability estimates using Lidstone smoothing [14] with $\epsilon = 1 \times 10^{-6}$, i.e.,

$$P(w_i|c_j) = \frac{t_{ij} + \epsilon}{\sum_i t_{ij} + \epsilon|\mathcal{V}|} \tag{15}$$

Where t_{ij} is the total number of occurrences of the word w_i in all documents belonging to class c_j. In practice, it is better to control for differing document sizes by normalizing each word vector representing a document, using the L2 norm. As such, we use these normalized word frequencies in our experiments. The smoothing above accounts for infrequent words that do not appear in the training set, but appear in the test set. Using $\epsilon << 1$ in Lidstone smoothing for word probabilities tends to work better for text classification than using Laplace smoothing ($\epsilon = 1$) [1].

For Pooling Multinomials, we set the *polarity level*, $r = 100$ — i.e., a positive word in the lexicon is considered 100 times more likely to appear in a positive document than a negative term. Section 4.5 presents a analysis of sensitivity to this parameter. We compare the different sentiment classification approaches on the datasets described in Section 4.1; where we pre-processed the data by filtering out stop-words and stemming words using the Porter stemmer [21]. We also eliminated infrequent words that appeared in less than 3 documents.

In order to evaluate the effect of varying amounts of training data, we generate learning curves averaged over multiple runs of 10-fold cross-validation. For Feature Supervision, we use the available training data at each point of the learning curve as the pool of unlabeled data. For *Lotus* and *Movies*, we used 10 runs of cross-validation; and for the more noisy *Politics* set we used 20 runs. The resulting learning curves are presented in Fig. 1, and the accuracies at the last point are summarized in Table 1. For clarity we only present the most relevant curves in the figure. In addition to accuracy, we also compared area under the ROC curve (AUC). The results on AUC are not presented here, as they are similar to results on accuracy.

Table 1: Comparing accuracy of different approaches to sentiment classification.

Model	*Lotus*	*Politics*	*Movies*
Lexical Classifier	68.23	55.20	63.40
Feature Supervision	57.93	46.19	57.59
Naïve Bayes	88.40	59.24	80.81
Linear Pooling	**91.21**	**63.61**	**81.42**
Log Pooling	88.42	60.04	80.00

The results clearly demonstrate that for all datasets, combining background knowledge with training examples via a Pooling Multinomials framework performs better than using each source of information in isolation. In particular, Linear Pooling performs the best, where all improvements in accuracy are statistically significant compared to other methods according to paired t-tests ($p < 0.05$). In general, Log Pooling is also effective, but the improvements are not as significant as taking the convex combination of distributions through Linear Pooling. In the rest of the paper, unless otherwise specified, we will use Pooling Multinomials to refer to Linear Pooling.

The learning curves show that, as expected, with very few training examples it is better to rely on only the background knowledge, than trying to estimate model parameters from a limited set of labels. However, with increasing amounts of labeled data, one can start building Naïve Bayes models that are better than the Lexical Classifier. Furthermore, combining both background knowledge with training data as in Pooling Multinomials is always a better alternative.

The impact of Pooling Multinomials is most dramatic when few labeled examples are available. However, a large number of labeled examples drawn from our data distribution should eventually capture the information being pro-

vided through background knowledge. We can see the beginning of this phenomenon in *Movies*, where the relative improvement of Pooling Multinomials over Naïve Bayes diminishes with an increasing number of training examples. However, in domains where examples have to be labeled by human experts, providing thousands of labels can be tedious. Being able to achieve high accuracies with a few training examples is a significant advantage is such situations. For example, in *Movies*, the accuracy of a Naïve Bayes model built using 800 training examples, can be achieved by Pooling Multinomials using about 300 examples. This can translate to a significant reduction in the labor-intensive process of human annotation. Using domain-independent resources such as the sentiment lexicon, also allows us to rapidly adapt to new domains by providing just a few domain-specific training examples to refine our background knowledge.

Using only the lexicon through the Lexical Classifier does not perform very well on our data sets. For *Politics* and *Movies* its accuracy is a little better than the base rate (majority-class rate) of 54% and 50% respectively; whereas for *Lotus*, its accuracy is below the base rate of 77%. The underlying assumption of the Lexical Classifier is that a document is positive if there are more positive lexicon terms than negative terms in a document. Apart from the fact that the lexicon does not cover all terms that may appear in our vocabulary, it also does not capture domain-specific connotations of terms. The Lexical Classifier also fails to account for the degree of positive and negative sentiment associated with each term. However, it is clear that these weaknesses of a pure Lexical Classifier can be overcome with learning from a few training examples. The effect of training on background knowledge is analyzed in more detail is Section 4.4.

We also observe that the Feature Supervision approach of Liu et al. performs quite poorly, doing worse than even the Lexical Classifier. This is counter to the previous results of Liu et al. [15]. However, in their study, they used very small lexicons of labeled features – as few as 5 and at most 30 per class. In contrast, in sentiment classification domains where large lexicons of thousands of word-class associations are readily available, a Lexical Classifier classifier performs better. Also, as pointed out before, the sentiment lexicon we use is general-purpose and needs to be refined for each domain. In Feature Supervision, learning from noisy examples labeled only by this domain-independent lexicon can clearly further deteriorate performance.

For our blog domains, we notice that the accuracy of sentiment classification is significantly higher for *Lotus* than for *Politics*. The *Lotus* posts originate from a small set of blogs, and there is almost a one-to-one correspondence between blogs and sentiment, i.e., each blogger either loves or hates *Lotus*. This conforms better with the assumptions of our generative model compared to the *Politics* posts, which originate from many disparate sources that have positive and negative commentary about different electoral candidates. As a result, our methods perform better at classifying sentiment in the *Lotus* blogs compared to the *Politics* blogs.

A good discussion on the challenges of sentiment classification for movie reviews can be found in [20] — we discuss some of the challenges of classifying sentiment specifically for blogs below.

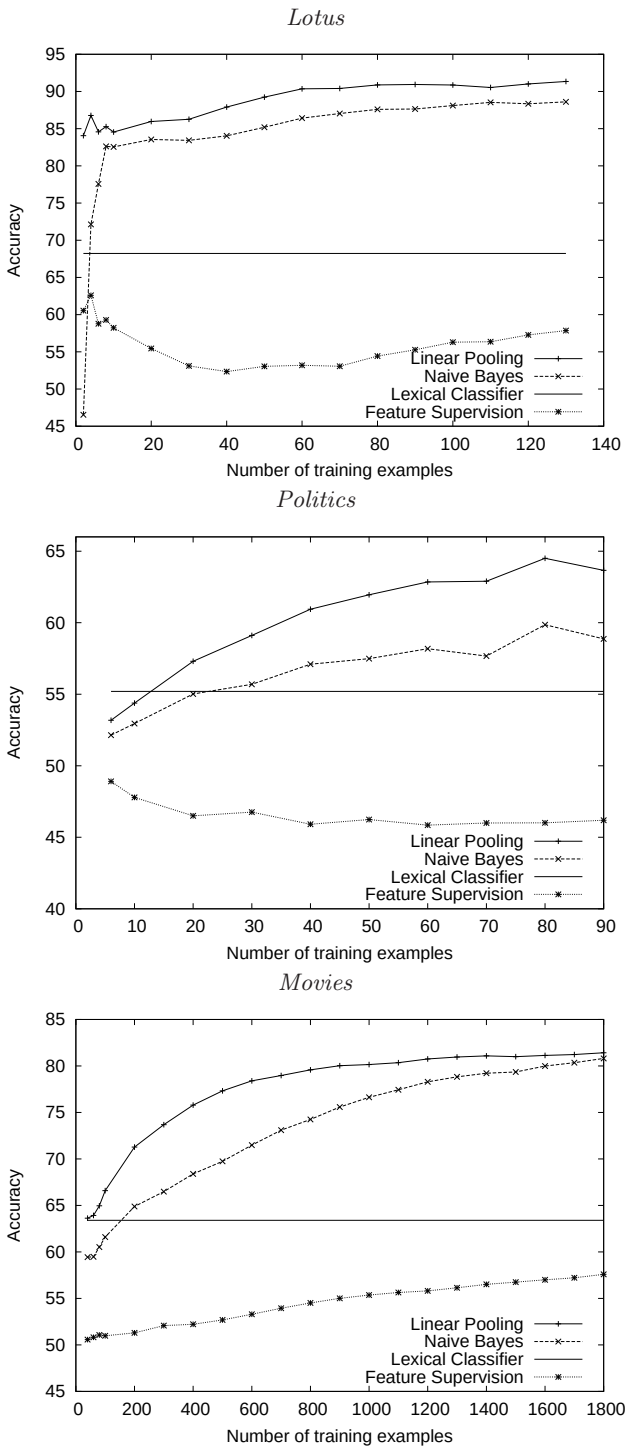

Figure 1: Comparing different approaches to sentiment classification.

4.3 Challenges in blog sentiment analysis

Labeling blog posts as positive or negative is very complex, even for humans. While efforts were made to focus solely on the post content when categorizing posts, it is still the case that comments, citations, and quotes from other sources are included in the main body of a post. When a

blog post's page becomes an area of discussion on a certain subject, labeling the entire page as positive or negative can be quite difficult. This is especially true for political blogs, where writers often make comparisons between multiple candidates, policies, or events. The issue of sentiment analysis is further complicated by the fact that bloggers often use jokes, anecdotes, and cultural references to illustrate their opinions, making the labeling task unclear for people unfamiliar with the relevant facts or references. This makes sentiment classification extremely difficult for most algorithms.

In related work, it has been observed that sentiment classifiers tend to perform better on more technical subjects than social or creative ones. For example, Turney [28] sees accuracies of 84% and 80% for reviews on automobiles and banks, respectively, and the same methods provide accuracies of only 66% for movies and 71% for travel destinations. Similarly, in our own results, political post classification tends to do far worse than labeling Lotus-focused posts. What is useful to note, however, is that our proposed methods still yield improved results on both domains.

4.4 Training examples versus background knowledge

Our results show that a generic lexicon is a very useful source of prior knowledge, that should not be dismissed, in lieu of training data. However, this raises the question of how exactly does supervised learning influence our background knowledge; and, can a single, comprehensive lexicon solve all our problems instead?

We claim that supervised learning refines our knowledge about the sentiment polarity of terms; and in particular, helps us learn domain-specific connotations. We can support this claim by examining the elements of our background knowledge that have been altered by training examples. Such insight can be easily gathered by comparing the Pooling Multinomials model with the background-knowledge model, and determining which lexicon terms have changed the most dramatically in sentiment. We can measure this directly, by computing the difference in log odds ratios of the conditional probabilities of words given the class. The log-odds of a word, $\log(P(w_i|+)/P(w_i|-))$, measures how indicative a word is of a document being in the positive class. One can compute this for all lexicon words in our final Pooling Multinomials model and in our background-knowledge model. Now the change in sentiment bias can be computed as the difference between log-odds with and without training. A positive value indicates that the corresponding word was up-weighted in terms of positive sentiment; and a negative value means that it was down-weighted in terms of positive sentiment.

Table 2 presents the top 10 up- and down-weighted terms for each domain. This gives us some insight into the domain-specificity of certain terms, which is not possible to encode into a single general-purpose lexicon. For example, words such as war, dark and complex can be associated with positive experiences in descriptions of movies, though they may be generally considered negative in other contexts. The down-weighting of positive lexicon terms, such as talent and promise for *Movies* is also consistent with the "thwarted expectation" narratives that Pang et al. [20] observed in this data.

While some of the positive or negative terms in Table 2 may seem counterintuitive, the up- and down-weighting represents the models' learning discussion patterns in specific domains. For example, truth has no negative connotation without a specific context, but is down-weighted in the politics-focused model. By exploring the posts that contain the term, one can see why – the seemingly positive term is used in sarcastic and accusatory posts, or within a negative context. For example, "spinning the truth", "transform a lie into a truth", or even specific contexts like "There is a lot of truth to Wright's sermons", a topic that usually contains negative sentiment towards Barack Obama.

Table 2: Terms from the lexicon that have been most up-weighted or down-weighted based on training examples.

Movies	up-weighted	war, flaw, social, dark, complex, anger, jar, edgy, capture, alien
	down-weighted	talent, reason, promise, save, fair, instinct, woo, adamant, redeem
Politics	up-weighted	oppose, service, margin, debate, incredible, war, serve, mean, contend, stick
	down-weighted	truth, subscribe, associate, connect, rally, liberal, definite, insist, accept, company
Lotus	up-weighted	cool, question, miss, theft, run, sorry, trick, social, silly, service
	down-weighted	attach, establish, origin, contain, correct, reason, select, sense, give, inform

4.5 Sensitivity analysis

The only parameter in Pooling Multinomials is the *polarity level* (r) used in the background knowledge model; and we explore its sensitivity to this parameter in this section. This *polarity level* is a measure of how much more likely it is for a positive term to occur in a positive context compared to a negative term, and vice-versa. We ran all tests on the *Movies* data, since it is the largest and provides results with the least variance. We measured the accuracy of our models for different values of r in the range 2 to 1000. The results, presented in Fig. 2, show that Pooling Multinomials is fairly robust to changes in this parameter — with accuracies only varying from 81.33 to 81.78. In practice, we set this parameter to 100 for all our experiments.

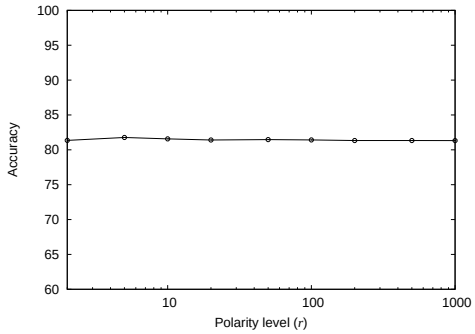

Figure 2: Evaluating sensitivity of Pooling Multinomials to the *polarity level* parameter.

5. RELATED WORK

Most work in sentiment analysis has focused on identifying positive or negative sentiment in text passages online. These studies can be broadly classified into two categories: knowledge-based approaches and learning-based approaches. Knowledge-based approaches primarily use linguistic models or other forms of background knowledge to classify the sentiment of passages. A large focus of this area is the use and generation of dictionaries capturing the sentiment of words. These methods range from manual approaches of developing domain-dependent lexicons [5] to semi-automated approaches [12, 33, 13, 32], and even an almost fully automated approach [28]. As observed by Ng et al. [17], most semi-automated approaches yield unsatisfactory lexicons, with either high coverage and low precision or vice versa. More recently, Pang et al. [20] successfully applied a machine learning approach to classifying sentiment for movie reviews. They cast the problem as a text classification task, using a bag-of-words representation of each movie review. They demonstrate that a learning approach performs better than simply counting the positive and negative sentiment terms using a hand-crafted dictionary. However, they do not consider combining such background lexical information with supervised learning, as we propose in this paper. Their results also suggest that using more sophisticated linguistic models, incorporating parts-of-speech and n-gram language models, do not improve over the simple unigram bag-of-words representation. In keeping with their findings, we also adopt a unigram text model. Pang et al. [19] extend their work, by first classifying sentences as *subjective* versus *objective*, and then classifying only the *subjective* sentences based on sentiment polarity. They demonstrate that by focusing only on the subjective sentences in each review they were able to improve the overall sentiment classification accuracy. Such a two-staged approach could also improve our results; however, for this paper we focus only on our advances in the polarity prediction stage. Durant and Smith [8] also apply text categorization to classification of political blog posts. Although their data is similar to ours, their task of identifying *left* versus *right* political alignment is quite different from our goal of identifying positive and negative sentiment. Using a Naïve Bayes classifier coupled with forward feature selection they were able to outperform SVMs. Given their success, we also use a Naïve Bayes approach. Wilson et al. [29] also formulate sentiment detection as a supervised learning task. However, instead of using just text classification, they focus on the construction of linguistic features, and train classifiers using Boostexter. Incorporating background knowledge, in terms of linguistic rules, in such classifiers is an interesting direction for future work.

Recently, there has been a growing interest in the use of background, prior or domain knowledge in supervised learning — including methods that use human-provided associations of features to particular classes. Most of this work has focused on using such prior class-bias of features to generate labeled examples that are then used for standard supervised learning. Schapire et al. [24] propose one such framework for boosting logistic regression, that uses hand-crafted rules generated from a list of relevant features to label *pseudo-examples*. They modify the boosting objective function to fit the training data, and the prior model based on these pseudo-examples. Provided with some features associated with each class, Wu and Srihari [31] assign labels to unlabeled documents, which are then used in conjunction with labeled examples to build a Weighted Margin Support Vector Machine. Unlike the above approaches, we use feature-class associations to directly construct a generative model. Dayanik et al. [6] explore several approaches to incorporating prior knowledge into Logistic Regression. In their study, human-annotated relevant features are given more ability to affect classification by assigning them a larger prior mode or variance than other features. Their approach of *mode-setting* is most closely related to ours. However, they report that this method was unreliable; since, it occasionally produced the best, but usually produced the worst results compared to other approaches. Druck et al. [7] incorporate prior knowledge through *labeled features*, which are used to directly constrain the model's predictions on unlabeled instances. Their Generalized Expectation criteria approach is applicable to any discriminative probabilistic model, and they demonstrate its utility specifically for multinomial logistic regression. Unlike their approach which uses only unlabeled instances, our method is supervised and uses background knowledge with labeled instances. However, we could extend our approach to also exploit unlabeled data, as discussed in Section 6. Sindhwani and Melville [26] propose an approach to incorporating labeled features and unlabeled documents within standard regularized least squares. In settings where labeled data is very limited and unlabeled data is abundant, their approach performs better than purely supervised and competing semi-supervised techniques. Incorporating background knowledge into learning has also been studied outside the context of text classification. Notably, Shavlik [25] has explored the use of prior knowledge in knowledge-based neural networks.

6. FUTURE WORK

Using background knowledge in supervised learning is one approach to reducing the burden of labeling many examples in the target domain. Another source of information that can be exploited, is labeled examples in related domains. For instance, a collection of reviews on software may capture a similar domain-dependent expression of sentiment that is also used in technology-blog discussions. Hence, one avenue for our future work is building better classifiers that exploit both background knowledge and labeled data from other domains. There has been a flurry of recent work in the area of transfer learning that could be applied to extend a background knowledge-based model to incorporate data from different domains. The fundamental challenge in such transfer learning is accounting for the training and test sets being from different distributions. Dai et al. [4] provide a solution for transfer learning in text classification using an EM-based Naïve Bayes classifier. Their solution first estimates the initial probabilities under a distribution $\mathcal{D}_l$ of the labeled data, and then uses the EM algorithm to revise the model for the test distribution $\mathcal{D}_u$, using unlabeled instances from the test set. This Naïve Bayes Transfer classifier can easily be added to the set of experts in Pooling Multinomials, thus combining both background knowledge and transfer learning in a single framework.

7. CONCLUSION

In this paper, we make two major contributions. First, we develop an effective framework for incorporating lexi-

cal knowledge in supervised learning for text categorization. Second, we successfully apply the developed approach to the task of sentiment classification — extending the state-of-the-art in the field which has focused primarily on using either background knowledge or supervised learning in isolation. Empirical results demonstrate that when provided with even a few training examples, we can combine background lexical information with supervised learning in our framework to produce better results than using a lexicon or the training data separately, as well as an approach to using a lexicon and unlabeled data in a semi-supervised setting. Though the primary focus of this paper is sentiment analysis, the approach developed is applicable to any text classification task in which some relevant background information is available. In the realm of blog analysis, such information may exist in various social and collaborative web-based tools like web tagging, folksonomies, or web directories. Exploiting such alternative sources of background knowledge in analyzing blogs provides interesting avenues for future work.

8. ACKNOWLEDGMENTS

We thank Yan Liu, Claudia Perlich, Vikas Sindhwani, Scott Spangler, and Ying Chen for insightful discussions.

9. REFERENCES

[1] R. Agrawal, R. J. B. Jr., and R. Srikant. Athena: Mining-based interactive management of text databases. In *Extending Database Technology*, 2000.

[2] Blogpulse: A service of nielsen buzzmetrics. http://www.blogpulse.com/.

[3] R. T. Clemen and R. L. Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19:187–203, 1999.

[4] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Transferring naive Bayes classifiers for text classification. In *AAAI*, 2007.

[5] S. Das and M. Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Asia Pacific Finance Association*, 2001.

[6] A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *SIGIR*, 2006.

[7] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

[8] K. T. Durant and M. D. Smith. *Advances in Web Mining and Web Usage Analysis*, chapter Predicting the Political Sentiment of Web Log Posts Using Supervised Machine Learning Techniques Coupled with Feature Selection. Springer, 2007.

[9] Extracting the main content from a webpage. http://w-shadow.com/blog/2008/01/25/extracting-the-main-content-from-a-webpage/.

[10] S. French. Group consensus probability distributions: A critical survey. In *Bayesian Statistics 2*, pages 183–197. North-Holland, 1985.

[11] C. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1:114–135, 1986.

[12] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

[13] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *COLING*, 2004.

[14] B. Liu. *Web Data Mining*. Springer, 2007.

[15] B. Liu, X. Li, W. S. Lee, and P. Yu. Text classification by labeling words. In *AAAI*, 2004.

[16] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *AAAI Workshop on Text Categorization*, 1998.

[17] V. Ng, S. Dasgupta, and S. M. N. Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *ACL*, 2006.

[18] K. Nigam. *Using Unlabeled Data to Improve Text Classification*. PhD thesis, Carnegie Mellon University, 2001.

[19] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, 2004.

[20] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, 2002.

[21] M. F. Porter. *An algorithm for suffix stripping*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[22] G. Ramakrishnan, A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya. Question answering via Bayesian inference on lexical relations. In *ACL Workshop on Multilingual Summarization and Question Answering*, 2003.

[23] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[24] R. E. Schapire, M. Rochery, M. G. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.

[25] J. Shavlik. A framework for combining symbolic and neural learning. In *Machine Learning*, 1992.

[26] V. Sindhwani and P. Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *ICDM*, 2008.

[27] S. Spangler, Y. Chen, L. Proctor, A. Lelescu, A. Behal, B. He, T. Griffin, A. Liu, B. Wade, and T. Davis. COBRA-Mining Web for Corporate Brand and Reputation Analysis. *IEEE International Conference on Web Intelligence*, 2007.

[28] P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *ACL*, 2002.

[29] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*, 2005.

[30] R. L. Winkler. The consensus of subjective probability distributions. *Management Science*, 15:361–375, 1968.

[31] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *KDD*, 2004.

[32] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP*, 2003.

[33] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *CIKM*, 2006.