For a few dollars less: Identifying review pages sans human labels

Luciano Barbosa
Dept. of Computer Science
University of Utah
Salt Lake City, UT 84112, USA.
lbarbosa@cs.utah.edu

Ravi Kumar **Bo Pang** **Andrew Tomkins**
Yahoo! Research
701 First Ave
Sunnyvale, CA 94089, USA.
{ravikumar,bopang,atomkins}@yahoo-inc.com

Abstract

We address the problem of large-scale automatic detection of online reviews *without* using any human labels. We propose an efficient method that combines two basic ideas: Building a classifier from a large number of noisy examples and using the structure of the website to enhance the performance of this classifier. Experiments suggest that our method is competitive against supervised learning methods that mandate expensive human effort.

1 Introduction

Shoppers are migrating to the web and online reviews are playing a critical role in affecting their shopping decisions, online and offline. According to two surveys published by comScore (2007) and Horrigan (2008), 81% of web users have done online research on a product at least once. Among readers of online reviews, more than 70% reported that the reviews had a significant influence on their purchases. Realizing this economic potential, search engines have been scrambling to cater to such user needs in innovative ways. For example, in response to a product-related query, a search engine might want to surface only review pages, perhaps via a "filter by" option, to the user. More ambitiously, they might want to dissect the reviews, segregate them into novice and expert judgments, distill sentiments, and present an aggregated "wisdom of the crowds" opinion to the user. Identifying review pages is the indispensable enabler to fulfill any such ambition; nonetheless, this problem does not seem to have been addressed at web scale before.

Detecting review webpages in a few, review-only websites is an easy, manually-doable task. A large fraction of the interesting review content, however, is present on pages outside such websites. This is where the task becomes challenging. Review pages might constitute a minority and can be buried in a multitude of ways among non-review pages — for instance, the movie review pages in `nytimes.com`, which are scattered among all news articles, or the product review pages in `amazon.com`, which are accessible from the product description page. An automatic and scalable method to identify reviews is thus a practical necessity for the next-generation search engines. The problem is actually more general than detecting reviews: it applies to detecting any "horizontal" category such as buying guides, forums, discussion boards, FAQs, etc.

Given the nature of these problems, it is tempting to use supervised classification. A formidable barrier is the labeling task itself since human labels need time and money. On the other hand, it is easier to generate an enormous number of low-quality labeled examples through purely automatic methods. This prompts the question: Can we do review detection by focusing just on the textual content of a large number of automatically obtained but low-quality labeled examples, perhaps also utilizing the site structure specific to each website? And how will it compare to the best supervised classification method? We address these questions in this paper.

Main contributions. We propose the first end-to-end method that can operate at web scale to efficiently detect review pages. Our method is based on using simple URL-based clues to automatically

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 494–502,
Boulder, Colorado, June 2009.

partition a large collection of webpages into two noisy classes: One that consists mostly of review webpages and another that consists of a mixture of some review but predominantly non-review webpages (more details in Section 4.2).

We analyze the use of a naive Bayes classifier in this noisy setting and present a simple algorithm for review page classification. We further enhance the performance of this classifier by incorporating information about the structure of the website that is manifested through the URLs of the webpages. We do this by partitioning the website into clusters of webpages, where the clustering delicately balances the information in the site-unaware labels provided by the classifier in the previous step and the site structure encoded in the URL tokens; a decision tree is used to accomplish this. Our classification method for noisily-labeled examples and the use of site-specific cues to improve upon a site-independent classifier are general techniques that may be applicable in other large-scale web analyses.

Experiments on 2000 hand-labeled webpages from 40 websites of varying sizes show that besides being computationally efficient, our human-label-free method not only outperforms those based on off-the-shelf subjectivity detection but also remains competitive against the state-of-the-art supervised text classification that relies on editorial labels.

2 Related work

The related work falls into roughly four categories: Document- and sentence-level subjectivity detection, sentiment analysis in the context of reviews, learning from noisy labeled examples, and exploiting site structure for classification.

Given the subjective nature of reviews, document-level subjectivity classification is closely related to our work. There have been a number of approaches proposed to address document-level subjectivity in news articles, weblogs, etc. (Yu and Hatzivassiloglou, 2003; Wiebe et al., 2004; Finn and Kushmerick, 2006; Ni et al., 2007; Stepinski and Mittal, 2007). Ng et al. (2006) experiment with review identification for known domains using datasets with clean labels (e.g., movie reviews vs. movie-related non-reviews), a setting different from that of ours. Pang and Lee (2008b) present a method on re-ranking documents that are web search results for a specific query (containing the word `review`) based on the subjective/objective distinction. Given the nature of the query, they implicitly detect reviews from unknown sources. But their re-ranking algorithm only applies to webpages known to be (roughly) related to the same narrow subject. Since the webpages in our datasets cover not only a diverse range of websites but also a diverse range of topics, their approach does not apply. To the best of our knowledge, there has been no work on identifying review pages at the scale and diversity we consider.

Subjectivity classification of within-document items, such as terms, has been an active line of research (Wiebe et al. (2004) present a survey). Identifying subjective sentences in a document via off-the-shelf packages is an alternative way of detecting reviews without (additional) human annotations. In particular, the OpinionFinder system (Riloff and Wiebe, 2003; Wiebe and Riloff, 2005) is a state-of-the-art knowledge-rich sentiment-analysis system. We will use it as one of our baselines and compare its performance with our methods.

There has been a great deal of previous work in sentiment analysis that worked with reviews, but they were typically restricted to using reviews extracted from one or two well-known sources, bypassing automatic review detection. Examples of such early work include (Turney, 2002; Pang et al., 2002; Dave et al., 2003; Hu and Liu, 2004; Popescu and Etzioni, 2005). See Pang and Lee (2008a) for a more comprehensive survey. Building a collection of diverse review webpages, not limited to one or two hosts, can better facilitate such research.

Learning from noisy examples has been studied for a long time in the learning theory community (Angluin and Laird, 1988). Learning naive Bayes classifiers from noisy data (either features or labels or both) was studied by Yang et al. (2003). Their focus, however, is to reconstruct the underlying conditional probability distributions from the observed noisy dataset. We, on the other hand, rely on the volume of labels to drown the noise. Along this spirit, Snow et al. (2008) show that obtaining multiple low-quality labels (through Mechanical Turk) can approach high-quality editorial labels. Unlike in their setting, we do not have multiple low-quality labels for the same URL. The extensive body of work in semi-supervised learning or learning from one class is also somewhat relevant to our work. A major difference is that they tend to work with small amount of clean, labeled data. In addition, many semi-supervised/transductive learning algorithms are not efficient for web-scale data.

Using site structure for web analysis tasks has been addressed in a variety of contexts. For example, Kening et al. (2005) exploit the structure of a website to improve classification. On a related note, co-training has also been used to utilize inter-page link information in addition to intra-page textual content: Blum and Mitchell (1998) use anchor texts pointing to a webpage as the alternative "view" of the page in the context of webpage classification. Their algorithm is largely site-unaware in that it does not explicitly exploit site structures. Utilizing site structures also has remote connections to wrapper induction, and there is extensive literature on this topic. Unfortunately, the methods in all of these work require human labeling, which is precisely what our work is trying to circumvent.

3 Methodology

In this section we describe our basic methodology for identifying review pages. Our method consists of two main steps. The first is to use a large amount of noisy training examples to learn a basic classifier for review webpages; we adapt a simple naive Bayes classifier for this purpose. The second is to improve the performance of this basic classifier by exploiting the website structure; we use a decision tree for this.

Let P be the set of all webpages. Let C_+ denote the *positive* class, i.e., the set of all review pages and let C_- denote the *negative* class, i.e., the set of all non-review pages. Each webpage p is exactly in one of C_+ or C_-, and is labeled $+1$ or -1 respectively.

3.1 Learning from large amounts of noisy data

Previous work using supervised or semi-supervised learning approaches for sentiment analysis assumes relatively high-quality labels that are produced either via human annotation or automatically generated through highly accurate rules (e.g., assigning positive or negative label to a review according to automatically extracted star ratings).

We examine a different scenario where we can automatically generate large amount of relatively low-quality labels. Section 4.2 describes the process in more detail, but briefly, in a collection of pages crawled from sites that are very likely to host reviews, those with the word `review` in their URLs are very likely to contain reviews (the noisy positive set $\widetilde{C}_+$) and the rest of the pages on those sites are less likely to contain reviews (the more noisy negative set $\widetilde{C}_-$). More formally, for a webpage p, suppose $\Pr[p \in C_+ \mid p \in \widetilde{C}_+] = \alpha$ and $\Pr[p \in C_+ \mid p \in \widetilde{C}_-] = \beta$, where $1 > \alpha \gtrsim \beta > 0$. Can we still learn something useful from $\widetilde{C}_+$ and $\widetilde{C}_-$ despite the labels being highly noisy?

The following analysis is based on a naive Bayes classifier. We chose naive Bayes classifier since the learning phase can easily be parallelized.

Given a webpage (or a document) p represented as a bag of features $\{f_i\}$, we wish to assign a class $\arg\max_{c \in \{C_+, C_-\}} \Pr[c \mid p]$ to this webpage. Naive Bayes classifiers assume f_i's to be conditionally independent and we have $\Pr[p \mid c] = \prod \Pr[f_i \mid c]$. Let $r_i = \Pr[f_i \mid C_+] / \Pr[f_i \mid C_-]$ denote the contribution of each feature towards classification, and $\mathrm{rc} = \Pr[C_+] / \Pr[C_-]$ denote the ratio of class priors. First note that

$$\log \frac{\Pr[C_+|p]}{\Pr[C_-|p]} = \log \left(\frac{\Pr[C_+]}{\Pr[C_-]} \cdot \frac{\Pr[p|C_+]}{\Pr[p|C_-]} \right)$$
$$= \log \left(\frac{\Pr[C_+]}{\Pr[C_-]} \cdot \prod r_i \right) = \log \mathrm{rc} + \sum \log r_i.$$

A webpage p receives label $+1$ iff $\Pr[C_+ \mid p] > \Pr[C_- \mid p]$, and by above, if and only if $\sum \log r_i > -\log \mathrm{rc}$.

When we do not have a reasonable estimate of $\Pr[C_+]$ and $\Pr[C_-]$, as in our setting, the best we can do is to assume $\mathrm{rc} = 1$. In this case, p receives label $+1$ if and only if $\sum \log r_i > 0$. Thus, a feature f_i with $\log r_i > 0$ has a positive contribution towards p being labeled $+1$; call f_i to be a "positive" feature. Typically we use relative-frequency estimation of $\Pr[c]$ and $\Pr[f_i \mid c]$ for $c \in \{C_+, C_-\}$. Now, how does the estimation from a dataset with noisy labels compare with the estimation from a dataset with clean labels?

To examine this, we calculate the following:

$$\Pr[f_i \mid \widetilde{C}_+] = \alpha \Pr[f_i \mid C_+] + (1-\alpha) \Pr[f_i \mid C_-],$$
$$\Pr[f_i \mid \widetilde{C}_-] = \beta \Pr[f_i \mid C_+] + (1-\beta) \Pr[f_i \mid C_-].$$

Let $\tilde{r}_i = \frac{\Pr[f_i|\widetilde{C}_+]}{\Pr[f_i|\widetilde{C}_-]} = \frac{\alpha r_i + (1-\alpha)}{\beta r_i + (1-\beta)}$. Clearly $\tilde{r}_i$ is monotonic but not linear in r_i. Furthermore, it is bounded:

$(1-\alpha)/(1-\beta) \leq \tilde{r}_i \leq \alpha/\beta$. However,

$\tilde{r}_i > 1 \iff \alpha r_i + (1-\alpha) > \beta r_i + (1-\beta)$ $\iff (\alpha-\beta) r_i > (\alpha-\beta) \iff r_i > 1$, where the last step used $\alpha > \beta$. Thus, the sign of $\log \tilde{r}_i$ is the same as that of $\log r_i$, i.e., a feature contributing positively to $\sum \log r_i$ will continue to contribute positively to $\sum \log \tilde{r}_i$ (although its magnitude is distorted) and vice versa.

The above analysis motivates an alternative model to naive Bayes. Instead of each feature f_i placing a weighted vote $\log \tilde{r}_i$ in the final decision, we trust only the sign of $\log \tilde{r}_i$, and let each feature f_i place a vote for the class C_+ (respectively, C_-) if $\log \tilde{r}_i > 0$ (respectively, $\log \tilde{r}_i < 0$). Intuitively, this model just compares the number of "positive" features and the number of "negative" features, ignoring the magnitude (since it is distorted anyway). This is precisely our algorithm: For a given threshold γ, the final label $\mathit{nbu}_\gamma(p)$ of a webpage p is given by

$$\mathit{nbu}_\gamma(p) = \operatorname{sgn}\left(\sum \operatorname{sgn}(\log \tilde{r}_i) - \gamma\right),$$

where sgn is the sign function. For comparison purposes, we also indicate the "weighted" version:

$$\mathit{nbw}_\gamma(p) = \operatorname{sgn}\left(\sum \log \tilde{r}_i - \gamma\right).$$

If $\gamma = 0$, we omit γ and use *nb* to denote a generic label assigned by any of the above algorithms.

Note that even though our discussions were for two-class and in particular, review classification, they are equally applicable to a wide range of classification tasks in large-scale web-content analysis. Our analysis of learning from automatically generated noisy examples is thus of independent interest.

3.2 Utilizing site structure

Can the structure of a website be exploited to improve the classification of webpages given by $\mathit{nb}(\cdot)$? While not all websites are well-organized, quite a number of them exhibit certain structure that makes it possible to identify large subsites that contain only review pages. Typically but not always this structure is manifested through the tokens in the URL corresponding to the webpage. For instance, the pattern `http://www.zagat.com/verticals/PropertyDetails.aspx?VID=a&R=b`, where `a,b` are numbers, is indicative of all webpages in `zagat.com` that are reviews of restaurants. In fact, we can think of this as a generalization of having the keyword `review` in the URL. Now, suppose we have an initial labeling $\mathit{nb}(p) \in \{\pm 1\}$ for each webpage p produced by a classifier (as in the previous section, or one that is trained on a small set of human annotated pages), can we further improve the labeling using the pattern in the URL structure?

It is not immediate how to best use the URL structure to identify the review subsites. First, URLs contain irrelevant information (e.g., the token `verticals` in the above example), thus clustering by simple cosine similarity may not discover the review subsites. Second, the subsite may not correspond to a subtree in the URL hierarchy, i.e., it is not reasonable to expect all the review URLs to share a common prefix. Third, the URLs contain a mixture of path components (e.g., `www.zagat.com/verticals/PropertyDetails.aspx`) and key-value pairs (e.g., `VID=a` and `R=b`) and hence each token (regardless of its position) in the URL could play a role in determining the review subsite. Furthermore, conjunction of presence/absence of certain tokens in the URL may best correspond to subsite membership. In light of these, we represent each URL (and hence the corresponding webpage) by a bag $\{g_i\}$ of tokens obtained from the URL. We perform a crude form of feature selection by dropping tokens that are either ubiquitous (occurring in more than 99% of URLs) or infrequent (occurring in fewer than 1% of URLs) in a website; neither yields useful information.

Our overall approach will be to use g_i's to partition P into clusters $\{C_i\}$ of webpages such that each cluster C_i is predominantly labeled as either review or non-review by $\mathit{nb}(\cdot)$. This automatically yields a new label $\mathit{cls}(p)$ for each page p, which is the majority label of the cluster of p:

$$\mathit{cls}(p) = \operatorname{sgn}\left(\sum_{q \in C(p)} \mathit{nb}(q)\right),$$

where $C(p)$ is the cluster of p. To this end, we use a decision tree classifier to build the clusters. This classifier will use the features $\{g_i\}$ and the target labels $\mathit{nb}(\cdot)$. The classifier is trained on all the webpages in the website and in the obtained decision tree, each leaf, consisting of pages with the same set of feature values leading down the path, corresponds to a cluster of webpages. Note that the clusters delicately balance the information in the site-unaware labels $\mathit{nb}(\cdot)$ and the site structure encoded in the URLs (given by g_i's). Thus the label $\mathit{cls}(p)$ can be thought of as a *smoothed* version of $\mathit{nb}(p)$.

Even though we can expect most clusters to be homogeneous (i.e., pure reviews or non-reviews), the above method can produce clusters that are inherently heterogeneous. This can happen if the website URLs are organized such that many subsites contain both review and non-review webpages. To take this into account, we propose the following hybrid approach that interpolates between the unsmoothed labels given by $\mathit{nb}(\cdot)$ and the smoothed labels given by $\mathit{cls}(\cdot)$. For a cluster C_i, the *discrepancy* $\mathrm{disc}(C_i) = \sum_{p \in C_i} [\mathit{cls}(p) \neq \mathit{nb}(p)]$; this quantity measures the number of disagreements between the majority label $\mathit{cls}(p)$ and the original label $\mathit{nb}(p)$ for each page p in the cluster. The decision tree guarantees $\mathrm{disc}(C_i) \leq |C_i|/2$. We call a cluster C_i to be δ-*homogeneous* if $\mathrm{disc}(C_i) \leq \delta |C_i|$, where $\delta \in [0, 1/2]$. For a fixed δ, the hybrid label of a webpage p is given by

$$\mathit{hyb}_\delta(p) = \begin{cases} \mathit{cls}(p) & \text{if } C(p) \text{ is } \delta\text{-homogeneous,} \\ \mathit{nb}(p) & \text{otherwise.} \end{cases}$$

Note that $\mathit{hyb}_{1/2}(p) = \mathit{cls}(p)$ and $\mathit{hyb}_0(p) = \mathit{nb}(p)$.

Note that in the above discussions, any clustering method that can incorporate the site-unaware labels $\mathit{nb}(\cdot)$ and the site-specific tokens in g_i's could have been used; off-the-shelf decision tree was merely a specific way to realize this.

4 Data

It is crucial for this study to create a dataset that is representative of a diverse range of websites that host reviews over different topics in different styles. We are not aware of any extensive index of online review websites and we do not want to restrict our study to a few well-known review aggregation websites (such as `yelp.com` or `zagat.com`) since this will not represent the less popular and more specialized ones. Instead, we utilized user-generated tags for webpages, available on social bookmarking websites such as `del.icio.us`.

We obtained (a sample of) a snapshot of URL–tag pairs from `del.icio.us`. We took the top one thousand sites with `review*` tags; these websites hopefully represent a broad coverage. We were able to crawl over nine hundred of these sites and the resulting collection of webpages served as the basis of the experiments in this paper. We refer to these websites (or the webpages from these sites, when it is clear from the context) as S_{all}.

4.1 Gold-standard test set

When the websites are as diverse as represented in S_{all}, there is no perfect automatic way to generate the ground truth labels. Thus we sampled a number of pages for human labeling as follows.

First, we set aside 40 sites as the test sites (S_{40}). In order to represent different types of websites (to the best we can), we sampled the 40 sites so that S_{40} covers different size ranges, since large-scale websites and small-scale websites are often quite different in style, topic, and content. We uniformly sampled 10 sites from each of the four size categories (roughly, sites with 100–5K, 5K–25K, 25K–100K, and 100K+ webpages)[1]. Indeed, S_{40} (as did S_{all}) covered a wide range of topics (e.g., games, books, restaurants, movies, music, and electronics) and styles (e.g., dedicated review sites, product sites that include user reviews, newspapers with movie review sections, religious sites hosting book reviews, and non-English review sites).

We then sampled 50 pages to be labeled from each site in S_{40}. Since there are some fairly large sites that have only a small number of review pages, a uniform sampling may yield no review webpages from those sites. To reflect the natural distribution on a website and to represent pages from both classes, the webpages were sampled in the following way. For each website in S_{40}, 25 pages were uniformly sampled (representing the natural distribution) and 25 pages were sampled from among "equivalence classes" based on URLs so that pages from each major URL pattern were represented. Here, each webpage in the site is represented by a URL signature containing the most frequent tokens that occur in the URLs in that site and all pages with the same signature form an equivalence class.

For our purposes, a webpage is considered a review if it contains significant amount of textual information expressing subjective opinions on or personal experiences with a given product / service. When in doubt, the guiding principle is whether

[1]As we do not want to waste human annotation on sites with no reviews at all, a quick pre-screening process eliminated candidate sites that did not seem to host any reviews.

a page can be a satisfactory result page for users searching for reviews. More specifically, the human annotation labeled each webpage, after thoroughly examining the content, with one of the following seven intuitive labels: "single" (contains exactly one review), "multiple" (concatenation of more than one review), "no" (clearly not a review page), "empty" (looks like a page that could contain reviews but had none), "login" (a valid user login needed to look at the content), "hub" (a pointer to one or more review pages), and "ambiguous" (border-line case, e.g., a webpage with a one line review). The first two labels were treated as $+1$ (i.e., reviews) and the last five labels were treated as -1 (i.e., non-reviews). Out of the 2000 pages, we obtained 578 pages labeled $+1$ and the 1422 pages labeled -1. On a pilot study using two human judges, we obtained 78% inter-judge agreement for the seven labels and 92% inter-judge agreement if we collapse the labels to ± 1. Percentages of reviews in our samples from different sites range from 14.6% to 93.9%.

Preprocessing for text-based analysis. We processed the crawled webpages using `lynx` to extract the text content. To discard templated content, which is an annoying issue in large-scale web processing, and HTML artifacts, we used the following preprocessing. First, the HTML tags `<p>`, `<br>`, `</tr>`, and `</td>` were interpreted as paragraph breaks, the '.' inside a paragraph was interpreted as a sentence break, and whitespace was used to tokenize words in a sentence. A sentence is considered "good" if it has at least seven alphabetic words and a paragraph is considered "good" if it has at least two good sentences. After extracting the text using `lynx`, only the good paragraphs were retained. This effectively removes most of the templated content (e.g., navigational phrases) and retains most of the "natural language" texts. Because of this preprocessing, 485 pages out of 2000 turned out to be empty and these were discarded (human labels on 97% of these empty pages were -1).

4.2 Dataset with noisy labels

As discussed in Section 3.1, our goal is to obtain a large noisy set of positive and negative labeled examples. We obtained these labels for the webpages in the training sites, S_{rest}, which is essentially $S_{\text{all}} \setminus S_{40}$. First, the URLs in S_{rest} were tokenized using a unigram model based on an English dictionary; this is so that strings such as `reviewoftheday` are properly interpreted.

$\widetilde{C}_+$: To be labeled $+1$, the path-component of the URL of the webpage has to contain the token `review`. Our assumption is that such pages are highly likely to be review pages. On a uniform sample of 100 such pages in S_{all}, 90% were found to be genuine reviews. Thus, we obtained a collection of webpages with slightly noisy positive labels.

$\widetilde{C}_-$: The rest of the pages in S_{rest} were labeled -1. Clearly this is a noisy negative set since not all pages containing reviews have `review` as part of their URLs (recall the example from `zagat.com`); thus many pages in $\widetilde{C}_-$ can still be reviews.

While the negative labels in S_{rest} are more noisy than the positive labels, we believe most of the non-review pages are in $\widetilde{C}_-$, and as most websites contain a significant number of non-review pages, the percentage of reviews in $\widetilde{C}_-$ is smaller than that in $\widetilde{C}_+$ (the assumption $\alpha \gtrneqq \beta$ in Section 3.1).

We collected all the paragraphs (as defined earlier) from both $\widetilde{C}_+$ and $\widetilde{C}_-$ separately. We eliminated duplicate paragraphs (this further mitigates the templates issue, especially for sites generated by content-management software), and trained a unigram language model as in Section 3.1.

5 Evaluations

The evaluations were conducted on the 1515 labeled (non-empty) pages in S_{40} described in Section 4.1. We report the accuracy (acc.) as well as precision (prec.), recall (rec.), and f-measure (fmeas.) for C_+.

Trivial baselines. Out of the 1515 labeled pages, 565 were labeled $+1$ and 950 were labeled -1. Table 1 summarizes the performance of baselines that always predict one of the classes and a baseline that randomly select a class according to the class distribution S_{40}. As we can see, the best accuracy is .63, the best f-measure is .54, and they cannot be achieved by the same baseline. Before present-

	acc.	prec.	rec.	fmeas.
always C_-	**.63**	-	0	-
always C_+	.37	.37	1	**.54**
random	.53	.37	.37	.37

Table 1: Trivial baseline performances.

ing the main results of our methods, we introduce a much stronger baseline that utilizes a knowledge-rich subjectivity detection package.

5.1 Using subjectivity detectors

This baseline is motivated by the fact that reviews often contain extensive subjective content. There are many existing techniques that detect subjectivity in text. OpinionFinder (`http://www.cs.pitt.edu/mpqa/opinionfinderrelease/`) is a well-known system that processes documents and automatically identifies subjective sentences in them. OpinionFinder uses two subjective sentence classifiers (Riloff and Wiebe, 2003; Wiebe and Riloff, 2005). The first (denoted *opf*$_A$) focuses on yielding the highest accuracy; the second (denoted *opf*$_B$) optimizes precision at the expense of recall. The methods underlying OpinionFinder incorporate extensive tools from linguistics (including, speech activity verbs, psychological verbs, FrameNet verbs and adjectives with frame "experiencer", among others) and machine learning. In terms of performance, previous work has shown that OpinionFinder is a challenging system to improve upon for review retrieval (Pang and Lee, 2008b). Computationally, OpinionFinder is very expensive and hence unattractive for large-scale webpage analysis (running OpinionFinder on 1515 pages took about five hours). Therefore, we also propose a light-weight subjectivity detection mechanism called *lwd*, which counts the number of opinion words in each sentence in the text. The opinion words (5403 of them) were obtained from an existing subjectivity lexicon (`http://www.cs.pitt.edu/mpqa`).

We ran both *opf*$_A$ and *opf*$_B$ on the tokenized text (running them on raw HTML produced worse results). Each sentence in the text was labeled subjective or objective. We experimented with two ways to label a document using sentence-level subjectivity labels. We labeled a document $+1$ if it contained at least k subjective sentences (denoted as *opf*$_\star(k)$, where $k > 0$ is the absolute threshold), or at least f fraction of its sentences were labeled subjective (denoted as *opf*$_\star(f)$, where $f \in (0, 1]$ is the relative threshold). We conducted exhaustive parameter search with both *opf*$_A$ and *opf*$_B$. For instance, the performances of *opf*$_A$ as a function of the thresholds, both absolute and relative, is shown in Figure 1. Table 2 summarizes the best performances of *opf*$_\star(k)$ (first two rows) and *opf*$_\star(f)$ (next two rows), in terms of accuracy and f-measure (bold-faced). Similarly, for *lwd*, we labeled a document $+1$ if at least k sentences have at least ℓ opinion words (denoted *lwd*(k, ℓ).) Table 2 once again shows the best performing parameters for both accuracy and f-measure for *lwd*. Our results indicate that a simple method such as *lwd* can come very close to a sophisticated system such as *opf*$_\star$.

	acc.	prec.	rec.	fmeas.
opf$_A(2)$	**.704**	.597	.634	.615
opf$_B(2)$	.659	.526	.857	**.652**
opf$_A(.17)$	**.652**	.529	.614	.568
opf$_B(.36)$	.636	.523	.797	**.632**
lwd$(1, 4)$	**.716**	.631	.572	.600
lwd$(1, 1)$	.666	.538	.740	**.623**

Table 2: Best performances of *opf*$_\star$ and *lwd* methods.

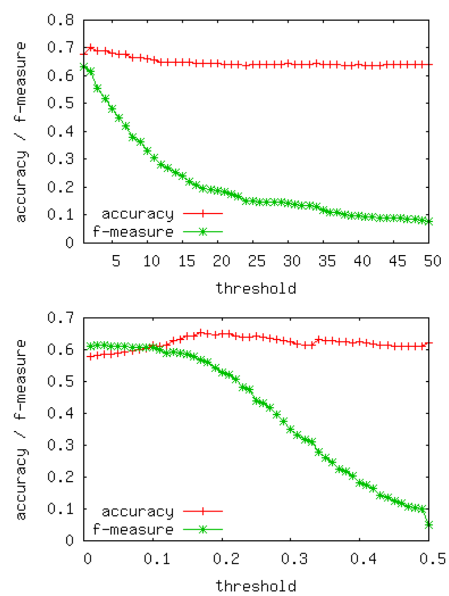

Figure 1: Performance of *opf*$_A$ as a function of thresholds: Absolute and relative.

5.2 Main results

As stated earlier, we do not have any prior knowledge about the value of γ and hence have to work with $\gamma = 0$. To investigate the implications of this assumption, we study the performance of *nbu*$_\gamma$ and *nbw*$_\gamma$ as a function of γ. The accuracy and f-measures are plotted in Figure 2. There are three

	acc.	prec.	rec.	fmeas.
nbu	.753	.652	.726	.687
cls	.756	.696	.616	.654
$hyb_{1/3}$	**.777**	.712	.674	**.693**

Table 3: Performance of our methods.

conclusions that can be drawn from this study: (i) The peak values of accuracy and f-measure are comparable for both nbu_γ and nbw_γ, (ii) at $\gamma = 0$, *nbu* is much better than *nbw*, in terms of both accuracy and f-measure, and (iii) the best performance of nbu_γ occurs at $\gamma \approx 0$. Given the difficulty of obtaining γ if one were to use nbw_γ, the above conclusions validate our intuition and the algorithm in Section 3.1.

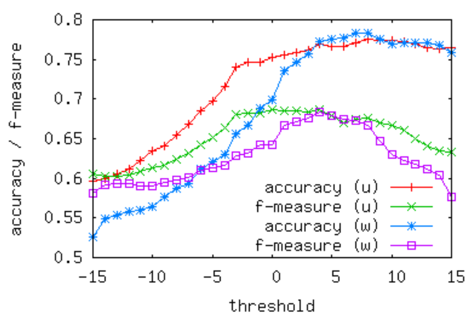

Figure 2: Performance as threshold changes: Comparing nbu_γ (marked as (u)) with nbw_γ (marked as (w)).

Table 3 shows the performance of the site-specific method outlined in Section 3.2. The clusters were generated using the unpruned J48 decision tree in Weka (`www.cs.waikato.ac.nz/ml/weka`). In our experiments, we set $\delta = 1/3$ as a natural choice for the hybrid method. As we see the performance of *nbu* is about 7% better than the best performance using a subjectivity-based method (in terms of accuracy). The performance of the smoothed labels (decision tree-based clustering) is comparable to that of *nbu*. However, the hybrid method $hyb_{1/3}$ yields an additional 3% relative improvement over *nbu*. Paired t-test over the accuracies for these 40 sites shows both $hyb_{1/3}$ and *nbu* to be statistically significantly better than the $opf_\star$ with best accuracy (with $p < 0.05$, $p < 0.005$, respectively), and $hyb_{1/3}$ to be statistically significantly better than *nbu* (with $p < 0.05$).

5.3 Cross-validation on S_{40}

While the main focus of our paper is to study how to detect reviews without human labels, we present cross validation results on S_{40} as a comparison point. The goal of this experiment is to get a sense of the best possible accuracy and f-measure numbers using labeled data and the state-of-the-art method for text classification, namely, SVMs. In other words, the performance numbers obtained through SVMs and cross-validation can be thought of as realistic "upper bounds" on the performance of content-based review detection. We used SVM^{light} (`svmlight.joachims.org`) for this purpose.

The cross-validation experiment was conducted as follows. We split the data by site to simulate the more realistic setting where pages in the test set do not necessarily come from a known site. Each fold consisted of one site from each size category; thus, 36 of the 40 sites in S_{40} were used for training and the remainder for testing. Over ten folds, the average performance was: accuracy .795, precision .759, recall .658, and f-measure .705.

Thus our methods in Section 3 come reasonably close to the "upper bound" given by SVMs and human-labeled data. In fact, while the supervised SVMs statistically significantly outperform *nbu*, they are statistically indistinguishable from $hyb_{1/3}$ via paired t-test over site-level accuracies.

6 Conclusions

In this paper we proposed an automatic method to perform efficient and large-scale detection of reviews. Our method is based on two principles: Building a classifier from a large number of noisy labeled examples and using the site structure to improve the performance of this classifier. Extensive experiments suggest that our method is competitive against supervised learning methods that depend on expensive human labels. There are several interesting avenues for future research, including improving the current method for exploiting the site structure. On a separate note, previous research has explicitly studied sentiment analysis as an application of transfer learning (Blitzer et al., 2007). Given the diverse range of topics present in our dataset, addressing topic-dependency is also an interesting future research direction.

References

Dana Angluin and Philip D. Laird. 1988. Learning from noisy examples. *Machine Learning*, 2(4):343–370.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of 45th ACL*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of 11th COLT*, pages 92–100.

comScore. 2007. Online consumer-generated reviews have significant impact on offline purchase behavior. Press Release, November. `http://www.comscore.com/press/release.asp?press=1928`.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of 12th WWW*, pages 519–528.

Aidan Finn and Nicholas Kushmerick. 2006. Learning to classify documents according to genre. *JASIST*, 7(5):1506–1518.

John A. Horrigan. 2008. Online shopping. Pew Internet & American Life Project Report.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of 19th AAAI*, pages 755–760.

Gao Kening, Yang Leiming, Zhang Bin, Chai Qiaozi, and Ma Anxiang. 2005. Automatic classification of web information based on site structure. In *Cyberworlds*, pages 552–558.

Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of 21st COLING/44th ACL Poster*, pages 611–618.

Xiaochuan Ni, Gui-Rong Xue, Xiao Ling, Yong Yu, and Qiang Yang. 2007. Exploring in the weblog space by detecting informative and affective articles. In *Proceedings of 16th WWW*, pages 281–290.

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

Bo Pang and Lillian Lee. 2008b. Using very simple statistics for review search: An exploration. In *Proceedings of 22nd COLING*. Poster.

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

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of HLT/EMNLP*, pages 339–346.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of EMNLP*, pages 105–112.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*.

Adam Stepinski and Vibhu Mittal. 2007. A fact/opinion classifier for news articles. In *Proceedings of 30th SIGIR*, pages 807–808.

Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th ACL*, pages 417–424.

Janyce M. Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of CICLing*, pages 486–497.

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

Yirong Yang, Yi Xia, Yun Chi, and Richard R. Muntz. 2003. Learning naive Bayes classifier from noisy data. Technical Report 56, UCLA.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP*, pages 129–136.