Semi-supervised Probabilistic Sentiment Analysis: Merging Labeled Sentences with Unlabeled Reviews to Identify Sentiment

Andrew Yates, Nazli Goharian
Information Retrieval Lab
Department of Computer Science
Georgetown University
{andrew,nazli}@ir.cs.georgetown.edu

Wai Gen Yee
Orbitz Worldwide
wyee@orbitz.com

ABSTRACT

Document level sentiment analysis, the task of determining whether the sentiment expressed in a document is positive or negative, is commonly performed by supervised methods. As with all supervised tasks, obtaining training data for these methods can be expensive and time-consuming. Some semi-supervised approaches have been proposed that rely on sentiment lexicons. We propose a novel supervised and a novel semi-supervised sentiment analysis method that are both based on a probabilistic graphical model, without requiring any lexicon. Our semi-supervised method takes advantage of the numerical ratings that are often included in online reviews (e.g., 4 out of 5 stars). While these numerical ratings are related to sentiment, they are noisy and hence, by themselves, they are an imperfect indicator of reviews' sentiments. We incorporate unlabeled user reviews as training data by treating the reviews' numerical ratings as sentiment labels while modeling the ratings' noisy nature. Our empirical results, utilizing a corpus of labeled sentences from hotel reviews and unlabeled hotel reviews with numerical ratings, show that treating reviews' ratings as noisy and utilizing them to augment a small amount of labeled sentences outperforms strong existing supervised and semi-supervised classification-based and lexicon-based approaches.

Keywords

Sentiment analysis, semi-supervised classification

INTRODUCTION

Sentiment analysis, the task of determining whether a text expresses positive sentiment, negative sentiment, or is neutral, is an active research topic with many applications. Sentiment analysis has grown in importance with the growth of social media. Many organizations benefit from a better understanding of the opinions and feelings expressed by groups and individuals towards an entity of interest. Sentiments are often expressed in online reviews of an entity, such as reviews of a restaurant, movie, hotel, etc. For example, potential patrons of a restaurant can benefit from knowing whether the sentiment expressed towards the restaurant is primarily positive or negative, whereas the restaurant's management can benefit from knowing the negative aspects its customers talk about. The former requires sentiment analysis at the document (i.e., review) level, whereas the latter requires sentiment analysis at the aspect level (i.e., knowledge of the sentiment towards an aspect of the restaurant, such as its service). By treating each sentence as a separate document, document level sentiment analysis can be directly performed at the sentence level.

Though aspect level sentiment analysis may sound more powerful at first, both types of analysis perform well in different situations. Document level sentiment analysis is useful for predicting the overall sentiment of documents, such as reviews, or of each sentence within a document. An example application of document level sentiment analysis is a hotel review database, in which the potential customers or management may query the database for sentences containing a keyword, such as "renovation," and view whether the sentences returned for a hotel are predominantly positive or negative in respect to that keyword. For example, as a hotel is undergoing a major renovation, the management might search for "renovation" to find out whether sentences that mention the term "renovation" are indicators of customers being more pleased with the newly renovated sections of the hotel or displeased with the extra noise and traffic accompanying the renovation process. Similarly, upon learning of ongoing renovations, a customer may search for "renovation" to learn whether she should avoid the hotel until the renovation is complete. This task is well-suited to document (i.e., review) level sentiment analysis as we are only concerned with knowing the sentiments expressed in the

ASIST 2013, November 1-6, 2013, Montreal, Quebec, Canada.
Copyright notice continues right here.

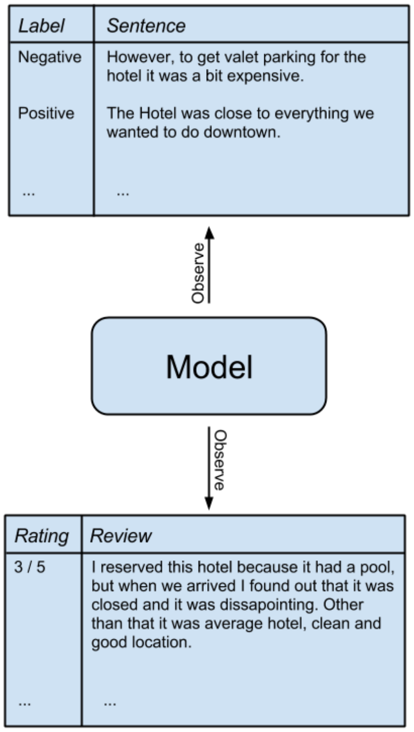
user review sentences returned by a search query; an information retrieval system returns the sentences of interest and a document-level sentiment analysis system determines the retrieved sentences' sentiments. Aspect level sentiment analysis, on the other hand, could perform this task only if every search query could be mapped directly to an aspect (i.e., characteristic or attribute). Using the above example, the keyword "renovation" may not be an aspect of hotel; hence, the user query would not be able to find the reviewers' sentiments about it by applying aspect level sentiment analysis. Aspect level sentiment analysis would produce incorrect results when the query cannot be mapped directly to an aspect. For example, a document level method would indicate that the sentence "*despite the major inconveniences caused by the ongoing renovation outside, the pool was clean*" is negative or neutral, as would a human annotator. Assuming that "renovation" is not an aspect, but "pool" is, an aspect level method would detect a positive sentiment towards "pool." It would be incorrect, however, to say that this sentence is expressing a positive sentiment only because of the sentiment towards the pool.

Supervised methods have often been applied to sentiment analysis at the document level, such as in (Paltoglou & Thelwall, 2010). By definition, supervised methods require training data, which is often difficult to obtain. In the context of sentiment analysis, training data consists of documents (or sentences) and labels indicating whether each document is positive, negative, or neutral. The documents must be manually labeled by human annotators and the process of doing so is time consuming. Furthermore, a recent work has found that a human annotator's accuracy does not increase with general annotation experience; instead, it increases with experience at a specific annotation task; that is, an annotator must spend time gaining experience at a specific task before achieving peak accuracy (Organisciak, Efron, Fenlon, & Senseney, 2012). As we describe later in our Related Work section, some efforts have proposed semi-supervised methods that combine supervised methods with a general sentiment lexicon (Melville, Ox, & Lawrence, 2009; L. Qiu, Zhang, Hu, & Zhao, 2009; Tan, Wang, & Cheng, 2008). These semi-supervised approaches require significantly less human effort, as they require less training data than a fully supervised approach; the sentiment lexicon they require (a list of words and their sentiments) can be built once and reused.

We propose a supervised probabilistic graphical model and a semi-supervised variant of the model; the semi-supervised variant outperforms other strong supervised and semi-supervised approaches. Our supervised model uses only sentences annotated for sentiment by humans ("labeled sentences") as training data. Our semi-supervised model combines labeled sentences with unlabeled user reviews and the numerical ratings commonly associated with user reviews (e.g., 3 out of 5 stars), as illustrated in Figure 1. This semi-supervised method utilizes unlabeled user reviews by treating the reviews' numerical ratings as sentiment labels while modeling the ratings' noisy nature. That is, the semi-supervised model treats the review ratings as sentiment labels that are less accurate than the sentiment labels provided by humans. We consider this to be semi-supervised because the user reviews are not labeled by human annotators. Furthermore, the numerical ratings found in reviews are often noisy and imperfect indicators of the sentiments expressed in reviews; their noisy nature must be modeled before they can reliably be used to infer sentiments. For example, a reviewer may give a hotel four stars while expressing negative sentiment in the review because she had high expectations (as might be the case with a luxury hotel). Similarly, a reviewer might give a hotel three stars while expressing both positive and negative sentiments in the review, such as in Figure 1.

Our contributions are

- Probabilistic Sentiment Analysis (PSA), a novel supervised method for document level sentiment analysis
- Semi-supervised Probabilistic Sentiment Analysis (S-PSA), a semi-supervised extension of PSA that improves PSA's performance by using both labeled sentences and unlabeled user reviews as training data
- an evaluation of our methods and comparison with several strong baselines



Label	*Sentence*
Negative	However, to get valet parking for the hotel it was a bit expensive.
Positive	The Hotel was close to everything we wanted to do downtown.
...	...

Rating	*Review*
3 / 5	I reserved this hotel because it had a pool, but when we arrived I found out that it was closed and it was dissapointing. Other than that it was average hotel, clean and good location.
...	...

Figure 1. Semi-supervised model combines labeled sentences with reviews that have numerical ratings

RELATED WORK

Much work has been done on sentiment analysis, which is the task of determining whether the sentiment expressed in a document (e.g., a review of a product or service) is positive, negative, or neutral. Sentiment analysis can be performed at the document (e.g., review) or aspect (e.g., product feature) level. Our method is focused on the document level. Surveys of sentiment analysis methods can be found in (Liu & Zhang, 2012) and (Pang & Lee, 2008). At a high level, most methodologies for sentiment analysis can be classified into a combination of three broad categories: *lexicon-based* approaches, *classification* approaches, and approaches that employ *probabilistic graphical models*. *Lexicon-based* approaches match words in a document against words in a sentiment lexicon to determine whether they are predominantly positive or negative. *Classification* approaches use training data, such as a sentiment lexicon or documents that have been labeled as positive or negative by humans, to train a classifier/model to classify documents as positive or negative. Lexicon-based and classification approaches often overlap (hybrid) because sentiment lexicons can be used to train a classifier.

Probabilistic graphical models, which are often used to identify the aspects talked about in a document (e.g., the speed, weight, and battery life of a laptop) and the sentiments expressed towards these aspects (e.g., a reviewer is pleased with the laptop's battery life and unhappy with its weight), identify aspects and sentiments by modeling how documents are written. For example, Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003), a common probabilistic graphical model used to model the topics expressed in documents, models documents as mixtures of topics and topics as distributions of words. While LDA's purpose is not to identify the sentiments expressed in documents, it shares similarities with many models with that purpose. The generative process of writing a document that LDA models is roughly: (1) choose a mixture of topics for the document to be about, (2) choose a topic for the next word to be about, (3) choose a word related to the chosen topic, (4) return to (2) to choose the next word, if the end of document is not reached. When applied to existing documents, LDA learns the word distribution associated with each topic and the topic distribution associated with each document.

Lexicon-based Approaches

(Hu & Liu, 2004) summarize users' opinions of products by identifying products' features and the sentiments towards these features in users' reviews. They identify a sentence's sentiment by determining the dominant sentiment polarity of the words in the sentence (i.e., determining whether more words are positive or negative). Hu & Liu build a sentiment lexicon to support this method. This lexicon is refined and expanded in Liu's later work; we will refer to it as *Liu's lexicon*. (Ding, Liu, & Yu, 2008) use Liu's lexicon with a scoring function that considers the distance between a feature and its sentiment words. (G. Qiu, Liu, Bu, & Chen, 2009) further improve Liu's lexicon by creating sentiment word extraction rules based on the relationships between sentiment words and the product features they are used with. (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011) create a sentiment lexicon that incorporates sentiments' intensities and defines the sentiment polarities of negated words. We use the current version of Liu's lexicon[1] (as of March 2013) and the method in (Hu & Liu, 2004) as one of our baselines; other methods that identify the sentiment towards a feature cannot be directly applied to our task of identifying sentiment at the document level.

Lexicons have been also used to create training data by identifying documents with a strong sentiment polarity. The sentiments attributed to these documents, by utilizing a lexicon, are used as training data for the classifier, which then predicts the sentiments of the remaining documents. This approach is used in (Tan et al., 2008), which we use as a baseline, and in (L. Qiu et al., 2009). A variant of this approach, in which lexicons are combined with other training data, is presented in (Melville et al., 2009).

Classification

The performance of various classifiers when applied to sentiment analysis was evaluated in (Pang, Lee, & Vaithyanathan, 2002); this study showed that Support Vector Machine (SVM) using Boolean unigram features performed the best (i.e., one feature per word is used to indicate whether the word is present). The impact of different feature types and feature weighting schemes was further explored in (Kim, Li, & Lee, 2009; Ng, Dasgupta, & Arifin, 2006; Paltoglou & Thelwall, 2010). These works found that unigrams weighted by TF-IDF and boolean unigrams generally perform well when compared with other features. (Paltoglou & Thelwall, 2010) found that a variant of TF-IDF that incorporates class information, Delta TF-IDF, performed better. We use a SVM with unigram features weighted by TF-IDF as a baseline; Delta TF-IDF does not directly apply to our task since it incorporates information for two classes whereas we have three classes (positive, negative, and neutral).

Probabilistic Graphical Models

Much work has applied probabilistic graphical models to the task of sentiment analysis with the purpose of identifying the aspects (or topics) and sentiments expressed in a document (Brody & Elhadad, 2010; Jo & Oh, 2011; Lin & He, 2009; Mei, Ling, Wondra, Su, & Zhai, 2007; Mukherjee & Liu, 2012a; Sauper, Haghighi, & Barzilay, 2011; Titov & McDonald, 2008). These methods model the relationships between words, aspects, sentiments, and documents. For example, the joint sentiment/topic model (JST) described in (Lin & He, 2009) models a document as a mixture of sentiment labels, each of which has a mixture

[1] http://cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

of topics; words are chosen from topics, which in turn are chosen from sentiments. Thus, each document may contain multiple sentiments and each sentiment may be associated with multiple aspects. For example, a restaurant review might contain positive and negative sentiments; the positive sentiment might be associated with the food and atmosphere, while the negative sentiment might be associated with the service.

All of these approaches differ from ours in that they identify sentiment at the aspect level, whereas our approach identifies sentiment at the document (review) level. Hence, we do not compare with any aspect level approaches because we model only the sentiment expressed in a document rather than modeling both the aspects and sentiment that are present. Our method can identify a sentence from a review as positive, negative, or neutral, whereas the aspect level methods identify the aspects present in a review and the sentiment expressed towards them. Both approaches are useful in different scenarios. We incorporate easily obtainable, yet noisy, star rating sentiments into our model to accurately identify sentiment at the document level. Although most probabilistic sentiment models are considered to be unsupervised, the knowledge about aspect and sentiment word distributions can be incorporated into many models' priors. The exception is (Mukherjee & Liu, 2012a), which uses category seed words to group the identified aspects into aspect categories. Our model is semi-supervised; that is, we use a combination of review sentences annotated for sentiment by humans and unlabeled reviews with only numerical ratings, to learn sentiments' word distributions.

Probabilistic graphical models have also been used to model attributes of reviews other than sentiment. While they do not model sentiment, these models share some similarities with the probabilistic models that do. (Moghaddam & Ester, 2011; Wang, Lu, & Zhai, 2010, 2011) model the relationships between aspects and a numeric rating (e.g., four out of five stars) associated with a review. (Lu & Zhai, 2008) model the relationships between experts' opinions and casually expressed opinions. Probabilistic graphical models have also been used to summarize users' feelings toward aspects of a service (Lu, Zhai, & Sundaresan, 2009) and to analyze comments posted in response to reviews (Mukherjee & Liu, 2012b).

METHODOLOGY

We first introduce our method, Probabilistic Sentiment Analysis (PSA), a supervised generative probabilistic model designed to identify the sentiment expressed in a document. In our approach, each sentence in a review is treated as a document. PSA is trained using sentences that are labeled with the sentiment expressed in the sentence. We then introduce our novel Semi-supervised Probabilistic Sentiment Analysis (S-PSA) method, an extension of PSA that can incorporate reviews' numerical ratings into its training.

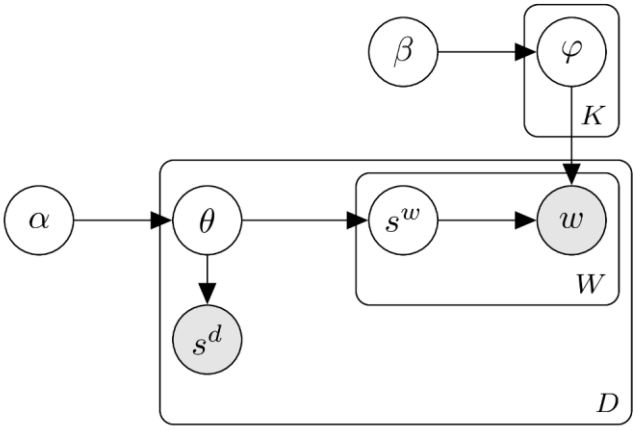


Figure 2. PSA (training phase)

Probabilistic Sentiment Analysis

Probabilistic Sentiment Analysis (PSA) models the sentiments present in a document similar to the way Latent Dirichlet Allocation (LDA) models the topics present in a document (Blei et al., 2003). The similarities and differences between LDA and PSA are discussed at the end of this section. PSA takes as input a collection of D documents $\{d_1, d_2, d_3, \dots, d_D\}$, each consisting of W word indexs $\{w_1, w_2, w_3, \dots, w_W\}$, where the value of each word index w_i corresponds to a word in the vocabulary V that is composed of $|V|$ distinct words. Let K be the number of sentiments that can be expressed; we set K=3 for positive, negative, and neutral sentiments. PSA's tasks are then (1) to learn the sentiment s^w of each word w in a document d, (2) to learn the overall sentiment s^d of document d, and (3) to learn each sentiment's word distribution φ_k. The word level and document level sentiments, s^w and s^d are related because both are drawn from a document level sentiment distribution θ.

While PSA's purpose is to identify the sentiment expressed in a document, not to generate documents, it is helpful to understand the generative process it models. Let $Dirichlet(\alpha)$ denote a Dirichlet distribution parameterized by the vector α and $Categorical(\varphi)$ denote a multinomial distribution with one trial parameterized by the vector φ. The generative process, also depicted in Figure 2, is then:

- For each sentiment k, choose a word distribution for the sentiment $\varphi_k \sim Dirichlet(\beta)$
- For each document d, choose a sentiment distribution for the document $\theta \sim Dirichlet(\alpha)$
- Choose an overall sentiment for the document $s^d \sim Categorical(\theta)$
- For each word w in the document
    - choose a sentiment for that word $s^w \sim Categorical(\theta)$
    - choose a word from the sentiment's word distribution $w \sim Categorical(\varphi_{s^w})$

Note that, as is the case with LDA, the Dirichlet distributions are symmetric. That is, each α_i in the hyperparameter vector α has the same value, as does each β_i in the hyperparameter vector β.

The document level sentiments s^d and the words in each document w are observed by the model. The word level sentiments s^W are latent variables because the model must infer the sentiment of each word from its document's sentiment distribution θ and the sentiment word distribution φ.

Given the hyperparameters α and β, the PSA's joint distribution is:

$$P(w, s^W, s^d, \theta, \varphi | \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i | \beta)$$

$$\prod_{j=1}^{D} P(\theta_j | \alpha)\, P(s_j^d | \theta_j) \prod_{k=1}^{W} P(s_{j,k}^W | \theta_j)\, P\left(w_{j,k} \middle| \varphi_{s_{j,k}^w}\right)$$

The first product is the product over the probability of each sentiment, the second product is the product over the probability of each document, and the last product is the product over the probability of each word in a document.

To train PSA, we employ uncollapsed Gibbs sampling (Casella & George, 1992), using the JAGS program (Plummer, 2003), to obtain the distributions of φ and θ and to obtain each s^W from a training set of documents. These training documents must have sentiment labels so that each s^d can be observed.

After training, the sentiment labels of a set of testing documents can be predicted. The document sentiment predictions are made by treating each s^d as a latent variable and treating φ as an observed variable (using the distribution of φ obtained in the training phase). After employing Gibbs sampling to estimate PSA's distributions on the new documents, the value of s^d and the distribution of θ predict the documents' sentiment labels.

If PSA did not model the document level sentiment s^d, its generative process would be equivalent to Latent Dirichlet Allocation's (LDA's) but with LDA's topics interpreted as sentiments. With the document level sentiment, PSA is equivalent to Labeled LDA with topics interpreted as sentiments (Ramage, Hall, Nallapati, & Manning, 2009). Incorporating the document level sentiment s^d is important in that it guides PSA to choose reasonable sentiments for each word within a document. Note that, LDA does not include a "document level topic"; if it did, then it would be analogous to PSA's document level sentiment. PSA also significantly differs from LDA in that it is supervised. PSA's training phase obtains a sentiment word distribution and its prediction phase predicts sentiments using that distribution; LDA has a single unsupervised phase.

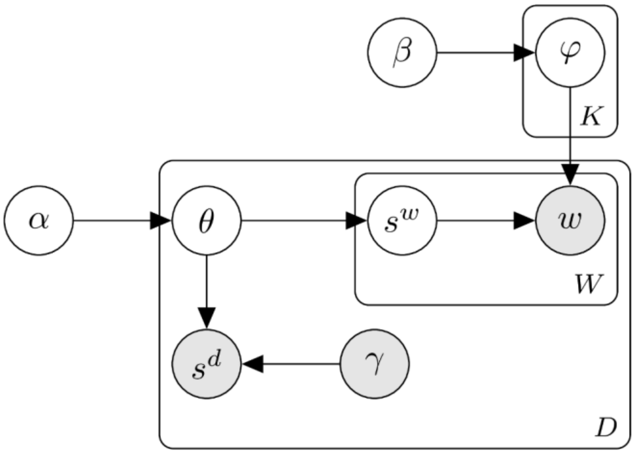


Figure 3. S-PSA (training phase)

Semi-Supervised Probabilistic Sentiment Analysis

Semi-supervised Probabilistic Sentiment Analysis (S-PSA) extends PSA to incorporate numerical review ratings in the training process in addition to utilizing sentences with sentiment labels. Note that the labeled sentences are sentences taken from reviews, whereas review ratings are associated with entire unlabeled reviews. To incorporate unlabeled reviews, PSA's generative process is modified so that a document's sentiment is drawn from a multinomial distribution with n=10 trials:

- Choose an overall sentiment for the document $s^d \sim Multinomial(\theta)$

The rest of the generative process remains the same. S-PSA is depicted in Figure 3. Note that the distribution that s^d is drawn from with PSA, the categorical distribution, is a special case of the multinomial distribution in which there is only n=1 trial.

In PSA, s^d is a scalar value that indicates the document's sentiment (e.g., "positive"), whereas in S-PSA, s^d is a vector that indicates the number of "successes" for each sentiment after performing n=10 draws (e.g., <negative=2, neutral=2, positive=6>). The number of successes for each sentiment is influenced by γ, a new document level variable.

Let i be an index into the vector s^d, so that $i = 0$ references the number of successes for the negative sentiment, $i = 1$ references the number of neutral successes, and $i = 2$ references the number of positive successes. Let $s^d[i]$ denote the number of successes at position i (e.g., $s^d[0]$ is the number of negative successes). Let s^{obs} be the document's observed sentiment, with $s^{obs} = 0$ being negative, 1 being neutral, and 2 being positive. s^d is then constructed as follows:

$$s^d[i] = \begin{cases} 10 - 2 * \gamma & if\ i = s^{obs} \\ \gamma & otherwise \end{cases}$$

Informally, the number of successes for the observed sentiment is $10 - 2 * \gamma$ and the number of successes for the other two sentiments is γ. For example, if $\gamma = 1$ and the document is positive, s^d = <negative=1, neutral=1, positive=8>. Likewise, if $\gamma = 0$ and the document is positive, s^d = <negative=0, neutral=0, positive=10>.

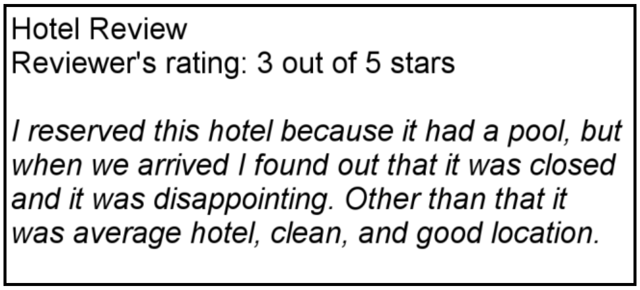
This formulation allows us to add noise to the document level sentiment observed by adjusting γ. Using a $\gamma > 0$ allows us to express that the observed sentiment is not perfectly reliable. While γ is technically document specific, we use only two values for γ in this work: one value for labeled sentences and one value for unlabeled reviews. Unlabeled reviews should have a larger γ than labeled sentences because a review's sentiment is estimated from its numerical rating, whereas labeled sentences are annotated by humans. S-PSA differs from Labeled LDA in that γ is document specific and in that "topics" (s^d) are drawn from a multinomial distribution.

S-PSA's joint distribution remains the same as PSA's. Similarly, S-PSA is trained in the same manner as PSA: each s^d is observed and φ is latent in the training phase, while each s^d is latent and φ is observed in the prediction phase.

EXPERIMENTAL SETUP

Data

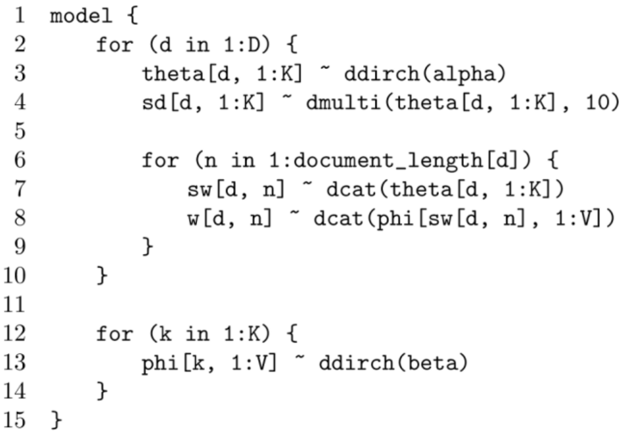
Our hotel review dataset consists of 3,565 reviews for 10 hotels in Chicago, Illinois, USA. The hotels were manually chosen by Orbitz employees to represent a wide range of hotel types. Each review consists of a rating on a 1 – 5 scale and review text. An example review is shown in Figure 4. We refer to this collection of hotel reviews as the *review collection*. The Punkt sentence tokenizer (Kiss & Strunk, 2006) was used to randomly select 240 sentences from the reviews. Three members of Orbitz's machine learning team manually labeled these 240 sentences to indicate whether the sentiment expressed in each was positive, negative, or neutral. We will refer to this group of 240 sentences as the *labeled sentences*. Note that while the reviews in the *review collection* have ratings provided by the reviewers, these ratings are noisy and thus may not be valid indicators of reviews' sentiments. For example, the review shown in Figure 4 has a neutral rating (3 out of 5 stars), but there are both negative and mildly positive sentiments expressed in the review. We treat a rating of 3 as a neutral sentiment, a rating lower than 3 as negative, and a rating higher than 3 as positive. A small subset of 48 sentences, randomly selected from the *labeled sentences,* was used to choose the parameters used by our models and the baselines.



Hotel Review
Reviewer's rating: 3 out of 5 stars

I reserved this hotel because it had a pool, but when we arrived I found out that it was closed and it was disappointing. Other than that it was average hotel, clean, and good location.

Figure 4. Example hotel review



```
model {
    for (d in 1:D) {
        theta[d, 1:K] ~ ddirch(alpha)
        sd[d, 1:K] ~ dmulti(theta[d, 1:K], 10)

        for (n in 1:document_length[d]) {
            sw[d, n] ~ dcat(theta[d, 1:K])
            w[d, n] ~ dcat(phi[sw[d, n], 1:V])
        }
    }

    for (k in 1:K) {
        phi[k, 1:V] ~ ddirch(beta)
    }
}
```

Figure 5. JAGS model for S-PSA

PSA and S-PSA

In our experiments, each α_i in the symmetric Dirichlet distribution α was set to 0.1, and each β_i in the symmetric Dirichlet distribution β was set to 0.2. These values were empirically chosen based on results with the subset of 48 sentences described in the previous section (Data).

PSA and S-PSA's distributions are obtained with Gibbs sampling, which is a Markov chain Monte Carlo algorithm (Andrieu, De Freitas, Doucet, & Jordan, 2003). Three Markov chains were used to perform the sampling. Each Markov chain was run for 5,000 steps after an initial burn-in period of 2,500 steps. The thinning parameter was set to 2, that is, every 2nd simulated draw was discarded. We performed the sampling with JAGS (Plummer, 2003). The JAGS model for S-PSA is shown in Figure 5.

Baselines

Our first baseline is a SVM^{light} (Joachims, 1999), with a linear kernel and unigram term features weighted by TF-IDF; this combination of features was found to perform best in (Kim et al., 2009; Pang et al., 2002). We refer to this baseline as *SVM*. In the S-PSA experiment, reviews from the *review collection* can be used with this method as training data by converting reviews' ratings to sentiments as previously explained in the Data section.

The second baseline is the current version of Liu's lexicon (as of March 2013) and the method in (Hu & Liu, 2004), as described in the related work section. We refer to this baseline as *Lexicon*.

Our third and final baseline is the approach described in (Tan et al., 2008) that combines a lexicon-based approach with a SVM classifier. With this method the sentiment of n% of the documents is predicted with a lexicon; these documents are then used to train a SVM classifier, which in turn is used to predict the sentiment of the remaining documents. We refer to this baseline as *Lexicon-SVM*. This method is unsupervised because the sentences' labels are not used; they are predicted using the lexicon.

In a semi-supervised variant of Lexicon-SVM, which we refer to as S-Lexicon-SVM, the training data is composed of both labeled sentences and unlabeled reviews whose ratings were predicted by the lexicon. That is, the SVM is trained using all of the labeled sentences and n% of the unlabeled reviews. We use n=40 when this method is used with only labeled sentences, namely Lexicon-SVM, and n=20 with the semi-supervised variant, namely S-Lexicon-SVM. These values of n were chosen to maximize the methods' performance on our data.

Metrics

We use precision, recall, and the F1 score as our metrics. These metrics are often used to evaluate methods for sentiment analysis and are also commonly used in the fields of data mining and information retrieval, among others. Precision is the fraction of a category's (sentiment's) predictions that are correct, while recall is the fraction of a category's correct instances that were predicted as belong to that category. In this work a category corresponds to a sentiment (i.e., the categories are positive, negative, and neutral). More formally, $precision = \frac{tp}{tp+fp}$ and $recall = \frac{tp}{tp+fn}$, where tp is the number of true positives, fp is the number of false positives, and fn is the number of false negatives. The F1 score, which we will refer to as F1, is the harmonic mean of precision and recall: $F1 = 2 * \frac{precision * recall}{precision + recall}$. When we report the average value of a metric, we do so using a macro average (i.e, we take the mean of the metric's value across all three categories).

Method	Average F1	Average Precision	Average Recall
PSA	0.55	0.56	0.55
Lexicon	0.53	0.67	0.44
SVM	0.53	0.51	0.55
Lexicon-SVM	0.34	0.32	0.36

Table 1: PSA results

Method	Average F1	Average Precision	Average Recall	F1 change from PSA
S-PSA	0.62	0.61	0.63	+13%
SVM	0.56	0.63	0.51	+2%
S-Lexicon-SVM	0.48	0.52	0.44	-13%

Table 2: S-PSA results

EVALUATION

We evaluate our methodologies, PSA and S-PSA, by comparing with the baselines described in the previous section, in respect to precision, recall, and F1 metrics. First, we present our results on PSA, Lexicon, SVM, and Lexicon-SVM, using the labeled sentences; we find that PSA's prediction quality is comparable to the best performing baseline. Next, we report results on S-PSA, SVM, and S-Lexicon-SVM, using both the labeled sentences and the review collection. We find that S-PSA significantly outperforms other methods in respect to F1. We then investigate the cause of S-PSA's performance improvement. Furthermore, we explore how S-PSA's performance changes when the number of reviews is reduced.

All results were obtained using 5-fold cross-validation with the supervised and semi-supervised methods (SVM, Lexicon-SVM, S-Lexicon-SVM, PSA, and S-PSA).

PSA results

We used PSA, Lexicon, SVM, and Lexicon-SVM to predict the sentiment of the labeled sentences described in the Data section. We used 5-fold cross-validation to train the supervised methods (PSA and SVM) on 80% of the data and test them on the remaining 20%. The results are shown in Table 1.

PSA, Lexicon, and SVM achieve close F1, with PSA performing approximately 4% better than the other two methods. Lexicon achieves a higher precision at the expense of a lower recall. Lexicon-SVM performs significantly worse, which is unsurprising given that it does not take advantage of training data as PSA and SVM do. Given the size of the *labeled sentence collection*, there may not be enough training data for Lexicon-SVM to learn from. Lexicon, which does not require any training and is thus not affected by the size of the training data, performs better than Lexicon-SVM.

S-PSA results

We used S-PSA, SVM, and S-Lexicon-SVM to predict the sentiment of the labeled sentences. This experiment differs from the previous (PSA) one, in that the methods in this experiment also use the reviews and numerical ratings from the *review collection* as training data.

Method	Average F1	Average Precision	Average Recall
PSA-Reviews	0.47	0.47	0.47
S-PSA-Sents-0	0.55	0.56	0.55
S-PSA-Sents-1	0.52	0.53	0.52
PSA	0.55	0.56	0.55
S-PSA	0.62	0.61	0.63

Table 3: Effect of modeling noise and including reviews in S-PSA's performance improvement

We previously described how SVM and S-Lexicon-SVM use the reviews (Baselines section), and how S-PSA uses the reviews (Methodology section).

We set S-PSA's $\gamma = 3$ for reviews and $\gamma = 1$ for labeled sentences. These values were empirically chosen based on the results obtained on a subset of the data set (described in the Data section). Intuitively, $\gamma = 1$ means that a document's label is reliable but not perfectly reliable. For example, if the document is positive, 10 draws from the document's sentiment distribution will result in 8 successes for positive, 1 success for negative, and 1 success for neutral. Similarly, $\gamma = 3$ intuitively means that a document's label is marginally reliable; for a positive document, 10 draws from the document's sentiment distribution results in 4 successes for positive, 3 successes for negative, and 3 successes for neutral. The outcomes with both values of γ correspond to what we believe the reliability of a document's label to be (i.e., review ratings are less reliable than sentences' sentiment labels).

S-Lexicon-SVM performs better than Lexicon-SVM (Table 1), but still performs worse than the other methods, as shown in Table 2. This is not surprising given that S-Lexicon-SVM is not using the reviews' numerical ratings. S-PSA and SVM, which are able to take advantage of the numerical ratings in the reviews, perform better. SVM performs better than in the previous experiment, with an F1 2% higher than PSA's. S-PSA performs significantly better than any other method performed in either experiment; its F1 is 13% higher than PSA's and 11% higher than SVM's.

S-PSA's performance improvement

As described in the methodology section, the main difference between PSA and S-PSA is that S-PSA models a document's sentiment distribution as a multinomial distribution with 10 trials and introduces γ to model noisy observations of a document's sentiment. Furthermore, another difference between the two is their training. S-PSA is trained on unlabeled reviews as well as on labeled sentences, whereas PSA is only trained on labeled sentences.

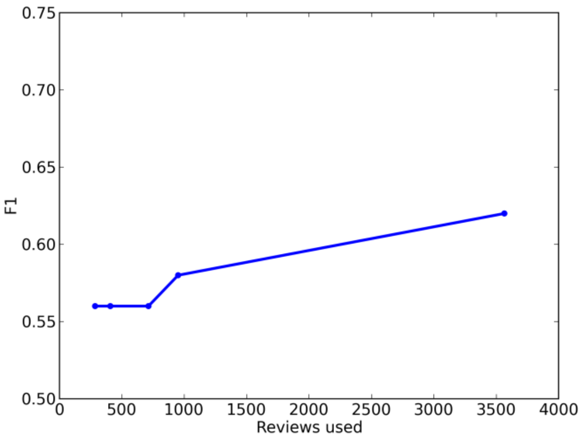


Figure 6. Impact of the number of reviews on S-PSA

We hypothesize that S-PSA's performance improvement over PSA is caused by both these differences. That is, we hypothesize that to outperform PSA, S-PSA must both use reviews and model the noisy nature of the review ratings. In this section we test this hypothesis by evaluating PSA and S-PSA in the following configurations:

- PSA-Reviews: Trained PSA on both the labeled sentences and unlabeled reviews. Each labeled sentence is treated as a document and each review is treated as a document.
- S-PSA-Sents-0: Trained S-PSA on only the labeled sentences with $\gamma = 0$.
- S-PSA-Sents-1: Trained S-PSA on only the labeled sentences with $\gamma = 1$.

The results for these runs are shown in Table 3. For ease in readability, we also include PSA's results from Table 1 and S-PSA's results from Table 2.

F1 decreases when we train PSA using both the labeled sentences and unlabeled reviews (PSA-Reviews); similarly, F1 decreases when we train S-PSA on only the labeled sentences (S-PSA-Sents-0 and S-PSA-Sents-1). These results support our hypothesis that S-PSA's performance improvement comes from both modeling noise and including reviews.

Performance impact of the review collection's size

We have shown in the previous experiments that (1) S-PSA performs better than the other methods and (2) S-PSA's performance improvement comes from both training on reviews and modeling the noise in the reviews. In this section we investigate the impact of the number of reviews used on S-PSA's performance. To do so, we ran S-PSA with 5-fold cross-validation as before, but we varied the number of reviews used for training. As before, we used $\gamma = 2$ for reviews and $\gamma = 1$ for labeled sentences.

The results are shown in Figure 6, which shows S-PSA's F1 as a function of the number of reviews. S-PSA's F1 score remains close to PSA's when it is trained on fewer than 1,000 reviews. S-PSA's F1 increases to 0.58 when 1,000 reviews are used and to 0.62 when all the reviews are used.

CONCLUSION

We have proposed two methods for performing document level sentiment analysis: PSA and S-PSA. PSA is supervised; it uses sentences with sentiment labels as training data. Our experiments show that PSA performs on par with strong document level sentiment analysis baselines. S-PSA is semi-supervised; it uses sentences with sentiment labels and unlabeled reviews with noisy numerical ratings as training data. S-PSA is able to learn from review ratings by treating them as sentiment labels that are less accurate than labeled sentences. PSA can be run on any text, whereas S-PSA's need for numerical ratings prevents it from generalizing to types of documents without numerical ratings. Future work could explore whether numerical ratings can be replaced with a confidence score from another sentiment analysis method; this would allow S-PSA to generalize to other types of documents.

We find that S-PSA outperforms PSA as well as strong supervised and semi-supervised baselines. An analysis of S-PSA's performance finds that incorporating reviews without modeling their noisy ratings is insufficient. This supports our hypothesis that S-PSA's improvement is caused by both including reviews and modeling their noisy ratings.

ACKNOWLEDGMENTS

This research is supported in part by National Science Foundation Award # CNS-1204347.

Orbitz provided the review collection and the Orbitz machine learning team manually labeled sentences from the review collection.

REFERENCES

Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning*, *50*(1-2), 5–43.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Brody, S., & Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '10)* (pp. 804–812).

Casella, G., & George, E. I. (1992). Explaining the Gibbs Sampler. *The American Statistician*, *46*(3), 167.

Ding, X., Liu, B., & Yu, P. (2008). A holistic lexicon-based approach to opinion mining. *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08)*.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04)* (pp. 168–177).

Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11)* (p. 815).

Joachims, T. (1999). Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning* (pp. 169–184). MIT Press.

Kim, J., Li, J., & Lee, J. (2009). Discovering the Discriminative Views : Measuring Term Weights for Sentiment Analysis. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP'09)* (pp. 253–261).

Kiss, T., & Strunk, J. (2006). Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, *32*(4), 485–525.

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09)*.

Liu, B., & Zhang, L. (2012). A Survey of Opinion Mining and Sentiment Analysis. In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data* (pp. 415–463). Boston, MA: Springer US.

Lu, Y., & Zhai, C. (2008). Opinion integration through semi-supervised topic modeling. *Proceedings of the 17th international conference on World Wide Web (WWW '08)* (p. 121). ACM Press.

Lu, Y., Zhai, C., & Sundaresan, N. (2009). Rated aspect summarization of short comments. *Proceedings of the 18th international conference on World wide web (WWW '09)* (pp. 131–140). Madrid, Spain: ACM.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic Sentiment Mixture : Modeling Facets and Opinions in Weblogs. *Proceedings of the 16th international conference on World Wide Web (WWW '07)*.

Melville, P., Ox, O., & Lawrence, R. D. (2009). Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09)*.

Moghaddam, S., & Ester, M. (2011). ILDA : Interdependent LDA Model for Learning Latent Aspects and their Ratings from Online Product Reviews Categories and Subject Descriptors. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11)* (pp. 665–674).

Mukherjee, A., & Liu, B. (2012a). Aspect extraction through semi-supervised modeling. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL '12)*.

Mukherjee, A., & Liu, B. (2012b). Modeling review comments. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*.

Ng, V., Dasgupta, S., & Arifin, S. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. *Proceedings of the COLING/ACL on Main conference poster sessions (COLING-ACL '06)* (pp. 611–618).

Organisciak, P., Efron, M., Fenlon, K., & Senseney, M. (2012). Evaluating rater quality and rating difficulty in online annotation activities. *ASIST '12* (Vol. 49, pp. 1–10).

Paltoglou, G., & Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)* (pp. 1386–1395).

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. (C. C. Aggarwal & C. Zhai, Eds.)*Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135. Now Publishers Inc.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP '02)*.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*.

Qiu, G., Liu, B., Bu, J., & Chen, C. (2009). Expanding Domain Sentiment Lexicon through Double Propagation. *Proceedings of the 21st international jont conference on Artifical intelligence (IJCAI '09)*.

Qiu, L., Zhang, W., Hu, C., & Zhao, K. (2009). SELC : A Self-Supervised Model for Sentiment Classification. *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09)* (pp. 929–936).

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA : A supervised topic model for credit attribution in multi-labeled corpora. *EMNLP '09* (pp. 248–256).

Sauper, C., Haghighi, A., & Barzilay, R. (2011). Content models with attitude. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, *37*(2), 267–307.

Tan, S., Wang, Y., & Cheng, X. (2008). Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)* (p. 743). New York, New York, USA: ACM Press.

Titov, I., & McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization. *Proceedings of ACL-08: HLT*.

Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: a rating regression approach. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10)* (pp. 783–792). Washington, DC, USA: ACM.

Wang, H., Lu, Y., & Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11)* (p. 618). New York, New York, USA: ACM Press.