Automatic Construction of a Context-Aware Sentiment Lexicon: An Optimization Approach

Yue Lu
UIUC Computer Science
201 N. Goodwin Avenue
Urbana, IL, USA
yuelu2@uiuc.edu

Malu Castellanos, Umeshwar Dayal
Intelligent Information Management Lab
HP Laboratories
Palo Alto, CA, USA
{malu.castellanos, umeshwar.dayal }@hp.com

ChengXiang Zhai
UIUC Computer Science
201 N. Goodwin Avenue
Urbana, IL, USA
czhai@cs.uiuc.edu

ABSTRACT

The explosion of Web opinion data has made essential the need for automatic tools to analyze and understand people's sentiments toward different topics. In most sentiment analysis applications, the sentiment lexicon plays a central role. However, it is well known that there is no universally optimal sentiment lexicon since the polarity of words is sensitive to the topic domain. Even worse, in the same domain the same word may indicate different polarities with respect to different aspects. For example, in a laptop review, "large" is negative for the battery aspect while being positive for the screen aspect. In this paper, we focus on the problem of learning a sentiment lexicon that is not only domain specific but also dependent on the aspect in context given an unlabeled opinionated text collection. We propose a novel optimization framework that provides a unified and principled way to combine different sources of information for learning such a context-dependent sentiment lexicon. Experiments on two data sets (hotel reviews and customer feedback surveys on printers) show that our approach can not only identify new sentiment words specific to the given domain but also determine the different polarities of a word depending on the aspect in context. In further quantitative evaluation, our method is proved to be effective in constructing a high quality lexicon by comparing with a human annotated gold standard. In addition, using the learned context-dependent sentiment lexicon improved the accuracy in an aspect-level sentiment classification task.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text Analysis; H.3.m [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

sentiment lexicon, sentiment analysis, opinion mining, optimization

1. INTRODUCTION

The advancement of Web 2.0 technologies has led to the explosive growth of online opinion data, which is becoming a valuable source for analyzing and understanding people's sentiments toward different topics. At the same time, it also brings the urgent need for automatic sentiment analysis tools. For this purpose, people have studied many sentiment analysis applications, such as opinion retrieval, opinion question answering, opinion mining, opinion summarization and sentiment classification. Essential to most of these applications is a comprehensive and high quality sentiment lexicon. Such a lexicon is not only necessary for sentiment analysis when no training data is available (in such a case, supervised learning would be infeasible), but is also useful for improving the effectiveness of any supervised learning approach to sentiment analysis through providing high quality sentiment features [2].

However, there is not a general-purpose sentiment lexicon that is optimal for all domains, because it is well known that sentiments of words are sensitive to the topic domain [19]. For example, "unpredictable" is negative in the electronics domain while being positive in the movie domain. Indeed, sentiment lexicons adapted to the particular domain or topic have been shown to improve task performance in a number of applications, including opinion retrieval [15, 9], and expression level sentiment classification [2]. Nevertheless, little attention has been paid to the further challenge that even in the same domain the same word may still indicate different polarities with respect to different aspects in context. For example, in laptop domain, "large" is negative for the battery aspect while being positive for the screen aspect.

In this paper, we focus on the problem of constructing a sentiment lexicon that is not only domain specific but also dependent on the aspect in context. Here, we use context-aware, context-dependent and aspect-dependent interchangeably, all referring to the expected output of a sentiment score assigned to each aspect and opinion word combination (e.g. BATTERY:large:-1). In particular, we are interested in methods generally applicable to any unlabeled opinionated corpus in any topical domain, so we make no assumption of the availability of human judged labels which are usually expensive to obtain in a new domain. Instead, we identify several sources of easy-to-collect information that are useful for determining the context-dependent sentiment of words. To solve the challenge that multiple signals come in different format and may even cause contradictions, we combine them through appropriate constraints in the objective function of a novel optimization framework, in which we search for optimal assignments of sentiment scores to aspect-opinion pairs that are most consistent with all the constraints. In this way, the optimization framework pro-

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

vides a unified and principled way to automatically construct a domain-specific aspect-dependent sentiment lexicon by consolidating multiple evidences from different sources.

More specifically, in the objective function, we combine the following four kinds of soft constraints, capturing four different sources of knowledge about sentiment, respectively: (1) constraints for sentiment priors which come from general-purpose sentiment lexicons, (2) constraints for overall sentiment ratings which provide the overall sentiments for all the words combined in the reviews, (3) constraints for similar sentiments which can be collected from synonyms in a thesaurus or from parsing the opinion collection with sentiment coherency assumption i.e. "and" rules as in linguistics heuristics, and (4) constraints for opposite sentiments which are from antonyms in a thesaurus or "but" rules in linguistics heuristics. These constraints cover most of the heuristics that have been exploited in existing work for inferring domain specific sentiments, and our method is the first to combine all those heuristics in a general and unified framework. More importantly, our constructed sentiment lexicon is not only domain specific but also aspect dependent.

To evaluate the effectiveness of our proposed framework, we conduct experiments on data sets in two different domains: hotel reviews and customer feedback surveys on printers. The results show that our approach can not only identify new sentiment words specific to the given domain (e.g. "private" is positive in hotel reviews; "compatible" is positive about printers) but also determine the different polarities of a word depending on the aspect in context (e.g. "huge room" v.s. "huge price" for hotels; "cheap ink" v.s. "cheap appearance" for printers). To further quantitatively evaluate the lexicon quality, we create a gold standard lexicon through human annotation, and our method is proved to be effective in constructing a high quality aspect-dependent sentiment lexicon. The results also demonstrate the advantage of combining multiple evidences over using any single evidence. Moreover, since the value of sentiment lexicons mostly lies in their usefulness in applications, we also study the performance of an aspect-level sentiment classification task by using the automatically constructed lexicon. The results show that using the context-dependent sentiment lexicon constructed by our optimization framework improves the sentiment classifier, compared with using baselines or a competitive method.

2. RELATED WORK

Sentiment analysis has attracted increasing attention recently. Sentiment lexicon plays an important role in most, if not all, sentiment analysis applications, including opinion retrieval [17], opinion question answering and summarization [3], opinion mining [21, 4] and unsupervised sentiment classification [7, 22, 19]. Even though supervised machine learning techniques have been shown to be effective for sentiment classification task (a detailed survey can be found in [18]), authors in [2] demonstrate that including features from sentiment lexicons boosts classification performance significantly.

However, manually creating a sentiment lexicon is a label intensive and error prone process; the coverage is also a concern. Thus, people studied the problem of creating a sentiment lexicon in an unsupervised manner [6, 19, 11, 8, 16, 14, 9, 5]. Moreover, there is no general-purpose sentiment lexicon that can work well for every domain or topic, because word sentiments are well known to be domain dependent [19]. Indeed, domain adapted sentiment lexicons have been shown to improve task performance in a number of applications, including opinion retrieval [15, 9], and expression level sentiment classification [2].

In those automatic methods, it is usually assumed that seed words with known polarity or a general-purpose sentiment lexicon is provided, whose polarity will be propagated to the unknown sentiment polarity of other words. Different heuristics as the propagation strategy have been proposed in existing work. Some are based on linguistic heuristics in the context [6, 11]. For example, two words linked by "but"-like conjunctions are most likely to be in opposite polarities, while conjunctions like "and" are evidences for words in the same polarity. Some works [16, 14] assume polarities of two words are correlated with their morphological relations and/or synonymy relations in thesaurus. Another popular line of methods, suggested by Turney [19], is to decide the polarity of a word or phrase by comparing whether it has a greater tendency to co-occur with the word "poor" (in a context window) or with the word "excellent" as measured by point-wise mutual information. Yet another kind of approaches exploit the association between words and expression-level or document-level sentiment [2, 12, 20]. There are also more recent works that combine more than one heuristic (i.e. linguistic heuristics and synonym/antonym rules) [4], but still in an ad-hoc rule-based manner which solves possible conflicting polarities by simple majority voting. To the best of our knowledge, there is no existing method that can combine all kinds of heuristics effectively in a unified framework, which is what we attempt to do in this work.

More importantly, although existing works try to learn word polarity in a specific domain, few of them consider the problem that even the same word in the same domain may indicate different polarities with respect to different aspects. A few studies attempted to generate word sentiment orientation dependent on aspects (or features) often as a side-product of their sentiment analysis problems [1, 12, 20, 10], but they have not directly evaluated the quality of the generated lexicon. Also, they all rely on a single source of information (document-level sentiment ratings or seed sentiment words), which is not sufficient as demonstrated in our experiments.

3. CONTEXT-DEPENDENT SENTIMENT LEXICON

We first define a general-purpose sentiment lexicon.

Definition (General-Purpose Sentiment Lexicon) A general-purpose sentiment lexicon L is a dictionary of opinion words where each word w is assigned a score representing the degree of sentiment. Conventionally, the sentiment score $L(w) \in [-1, 1]$; and in many cases it is binary, i.e. either $+1$ (positive) or -1 (negative).

Our goal is to automatically construct a context-dependent sentiment lexicon, which can be used to supplement the general sentiment lexicon and provide more accurate context-dependent sentiment information for different applications, such as sentiment classification, opinion summarization, opinion retrieval and so on.

To construct a context-dependent sentiment lexicon, we

assume that a set of aspects are given: $A = \{A_1, A_2, \cdots, A_k\}$, where each aspect is defined as follows:

Definition (Aspect) An aspect A_i is a set of terms characterizing a subtopic or a theme in a given domain, which can be features of products or attributes of services. For example, words such as "breakfast", "restaurant", and "pizza" can characterize the aspect about food in hotel reviews. We denote an aspect by $A_i = \{a;\ f(a) = i\}$, where $f(a)$ is a mapping function from a word a to its aspect index i.

Such aspects can be obtained through domain experts manual effort, or unsupervised automatic methods (e.g. [10]), or automatic methods with specified user interests as minimal human supervision (e.g. [13]). It is not our focus to find those aspects. Instead, assuming the availability of aspects, our problem is to automatically construct a context-dependent sentiment lexicon, defined as follows:

Definition (Context-Dependent Sentiment Lexicon) A context-dependent sentiment lexicon L_c is a dictionary of opinion words conditioned on different aspects of the given domain. Each entry in L_c is a pair of aspect A_i and opinion word w, and it is assigned a score representing the positive or negative sentiment it is expressing. $L_c(A_i, w) \in [-1, 1]$.

Our general idea of constructing such a lexicon is to leverage many naturally available resources, which we will discuss in detail in the next section.

4. MULTIPLE SOURCES OF USEFUL SIGNALS

We do not make any assumption about the availability of human judged labels because they are usually expensive to obtain in a new topic domain. Nevertheless, we identify several kinds of easy-to-collect information that are helpful signals in determining the context-dependent sentiments of words. Here we summarize and categorize different sources of signals.

- **General-purpose sentiment lexicon**, which contains words that are almost always positive or negative in any domain, such as "excellent" and "bad". This lexicon provides high confidence but low coverage sentiments.
- **Overall sentiment rating**, i.e. sentiment rating/score at the document level. In many cases, each opinionated text comes with an overall sentiment rating from the user, such as in TripAdvisor[1], Epinions[2], and Amazon[3] reviews. Such kind of data is abundant on the Web. For example, there are more than 40 million travel-related reviews on TripAdvisor, and millions of reviews on millions of products from Epinions. The intuition is that the overall rating conveys some information about the sentiment expressed in the text. For example, it is very unlikely that a user uses all negative words in the text while giving an overall rating of 5 stars.
- **Thesaurus**, which contains synonym and antonym information, such as WordNet[4]. For example, we may not know whether "large" is positive or negative for the screen aspect in laptop reviews, but we know it should be very similar to "big" and very different from "tiny". Then if we have some other evidences about the polarity of "big" or "tiny", we can better infer the polarity of "large".
- **Linguistic heuristics**
  - "and" rule: Clauses that are connected with "and"-like conjunctives usually express the same sentiment polarity. For example, "battery lasts long and screen size is large" implies that "long" for "battery" and "large" for "screen size" are of the same polarity. Other terms include: as well as, likewise.
  - "but" rule: Clauses that are connected with "but"-like conjunctives usually express the opposite sentiment polarity. For example, "battery lasts long but screen size is tiny" indicates that "long" for "battery" and "tiny" for "screen size" are of the opposite polarity. Other terms include: however, nevertheless, though, although, except that, except for, besides, with the exception of, despite, in spite of.
  - "negation" rule: Negation words such as "no", "not", and "never" reverse the sentiment of the opinion word in the same clause. For instance, "not happy" should have opposite sentiment as "happy".

[1] http://www.tripadvisor.com
[2] http://www.epinions.com
[3] http://www.amazon.com
[4] http://wordnet.princeton.edu

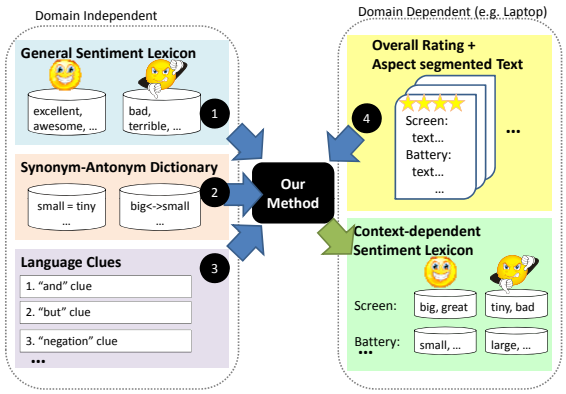

Figure 1: Problem Overview

These categories cover most of the heuristics used in existing works of learning domain specific sentiment lexicon, but no previous work has combined all these sources of signals. Since the information from any single source can be sparse, it would be helpful if we can combine the signals from multiple sources effectively. To this end, we propose to combine all the information from different signals and learn a context-dependent sentiment lexicon, as illustrated in Figure 1. The idea is that when the signal from one source is not available or not confident enough, we can still refer to other signals to fill the gap. In the next section, we propose a novel optimization framework to effectively combine different kinds of signals in a unified way.

5. AN OPTIMIZATION FRAMEWORK

Due to the fact that the different signals come in different format, it is not clear how to combine them in a unified way. Moreover, there can be contradictory signals from different sources, which we also need to deal with. We first discuss how we generate all the candidate lexicon entries which form

the search space for the optimization problem, and then define components in the objective function to capture various constraints. Finally, we show how we transform the proposed optimization framework into a linear programming problem which has efficient solutions and locally optimal solutions are also guaranteed to be globally optimal.

5.1 Generation of Candidate Lexicon Entries

The goal of this step is to tag the text collection with aspects and extract candidate opinion words to be paired with the aspects. After that, the pairs serve as entries in the context-dependent sentiment lexicon which are going to be assigned with polarity scores by our optimization method.

It is common to use each sentence as a tagging unit. But it is often the case, especially in online reviews, that one sentence covers different aspects in several subsentences or clauses; in addition, one clause can express sentiment of different polarity than other clauses in the same sentence. Thus, we choose to use clauses as units instead of sentences; this allows us to associate potential opinions words with the aspects more accurately. We employ the Stanford Parser to do the sentence splitting and to parse sentences into syntactic tree structures. Then we use the subtrees tagged as "simple declarative clause", as candidate clauses. We also manually set a few rules to merge fragmental clauses into longer and more meaningful ones.

After that, we can now tag each clause s with the corresponding aspects. Since we already have a set of defined aspects A in the form of word clusters, we take the straightforward way that is to tag the clause with the aspects whose word cluster overlaps with the words in the clause. Now we have the opinionated text segmented into clauses which are tagged with the corresponding aspects. An example sentence is as follows, where two clauses are in brackets: the first clause is tagged with the SERVICE aspect because "check in" appears in the word cluster of SERVICE; similarly, the second clause is tagged with the FOOD aspect.

```
[The (check in):SERVICE is very smooth] and
[the (restaurant):FOOD is the best].
```

Finally, the other non-aspect and non-stop words in each clause are considered potential opinion words in the context of the tagged aspects. In the previous example, we will extract the pairs (SERVICE, very) and (SERVICE, smooth) from the first clause and (FOOD, best) from the second clause. If one clause has been tagged with more than one aspect, we will pair the potential opinion words with each aspect. It is possible to employ other aspect segmentation and tagging techniques to extract the candidate pairs, but we choose a simple and trackable approach here in order to focus on the next step of sentiment learning.

5.2 Constraints in the Objective Function

We propose to formulate this as an optimization problem. Basically, we will be searching for a sentiment score assignment to candidate lexicon entries that optimizes the objective function. To design the objective function, there will be constraints defined from different sources of information so that the optimal solution to the objective captures the intuitions behind different evidences.

Formally, suppose we are provided with a collection of m opinionated text data (or reviews for short) $D = \{d_1, d_2, ..., d_m\}$ in a given domain, k defined aspects and n candidate lexicon entries extracted from the previous step, i.e. n is the number of aspect-opinion pairs. Our goal is to compute S, a $n \times 1$ vector, where each $S_j \in [-1, 1]$ indicates the sentiment score of the aspect-opinion pair j in the given domain. For convenience, let a_j denote the aspect of j, w_j the opinion word in pair j. Basically, S_j is a concise representation of an entry in the context-dependent sentiment lexicon as defined in Section 3, i.e. $S_j = L_c(a_j, w_j)$.

Constraints for Sentiment Prior: Given an aspect-opinion pair j, if we do not have any clue about the polarity of word w_j in the special context of aspect a_j , a natural guess is w_j's sentiment score in a general-purpose sentiment lexicon (if it is in there), which should give us good prior information.

Provided with a general-purpose sentiment lexicon L, we define two $n \times 1$ vectors G and I^G: for each pair j, we set $G_j = L(w_j)$ and $I^G_j = 1$ if w_j exists in L; otherwise, $G_j = 0$ and $I^G_j = 0$. Basically, I^G_j is an indicator as whether the word w_j has prior sentiment score or not while G_j is the score if there is one available. Now we introduce the first part of our objective function

$$\text{minimize} \left\{ \sum_{j=1}^{n} I^G_j \, |S_j - G_j| \right\} \tag{1}$$

This component in the objective function favors a context-dependent sentiment score assignment of S that is closest to the general-purpose sentiment lexicon, i.e. G.

Constraints for Overall Sentiment Ratings: Unlike the general-purpose sentiment lexicon that provides the prior sentiment information of words, overall sentiment ratings only represent the sentiment score at the document level. Nevertheless, it is usually assumed that the overall sentiment rating are positively correlated with the sentiments of the words in the document, which has been validated in some existing work [12, 20].

We define O as a $m \times 1$ vector, where O_i is the overall sentiment rating of the review text d_i normalized to $[-1, 1]$. Let $f(d_i, S)$ be a sentiment prediction function that outputs a sentiment score based on the review text d_i and our context-dependent sentiment lexicon S. Then we want the sentiment score calculated from our lexicon to be close to the overall sentiment rating which is observed, i.e.

$$\text{minimize} \left\{ \sum_{i=1}^{m} I^O_i \, |f(d_i, S) - O_i| \right\} \tag{2}$$

where I^O_i is again an indicator as whether O_i is defined, which offers flexibility in our framework, because not all reviews have overall sentiment rating available. Here, we choose a simple but commonly-used sentiment prediction function: averaging the sentiment scores of aspect-opinion pairs appearing in the review text based on our context-dependent sentiment lexicon. Formally, let X bs a $m \times n$ co-occurrence matrix, where each X_i is a $1 \times n$ vector representing the unigram language model of review d_i in terms of aspect-opinion pairs. In other words, X_{ij} is the number of times that the particular pair j occurs in review d_i divided by the total number of pairs in review d_i. We also take into account the "negation" rules here: If there are any negation words in the same clause, we replace the count of this occurrence from 1 to -1 when estimating X_{ij}. Then, replacing $f(d_i, S)$ with $\sum_{j=1}^{n} X_{ij} S_j$ in term (2), we have the

following term as the second part in the objective function

$$\text{minimize} \left\{ \sum_{i=1}^{m} I_i^O \left| \sum_{j=1}^{n} X_{ij} S_j - O_i \right| \right\} \quad (3)$$

This term (3) is basically a linear regression formulation where we are looking for a solution for the unknown variables S by minimizing the distance between the observed values of the dependent variable O and the predicted values which are based on the independent variables X. matrix).

Constraints for Similar Sentiments: We can collect evidences about similar sentiments from different sources. Consider any two aspect-opinion pairs j and k on the same aspect (i.e. $a_j = a_k$), if w_j and w_k appear as synonyms in the thesaurus, or if the pairs j and k are often concatenated with conjunctives like "and" in the corpus, we can infer that their sentiments tend to be similar.

To formalize this intuition, we define A, a $n \times n$ matrix, where $A_{jk} \in [0, 1]$ denotes our confidence about pairs j and k having similar sentiments. A simple way to construct the matrix A is to set A_{jk} to 1 if $a_j = a_k$ and either w_j, w_k are synonyms in the thesaurus or pairs j, k are conjuncted by "and" linguistic heuristic in the review text for a minimal number of times; while leaving the other elements as zeros. A more sophisticated way is to use a graded confidence score in A instead of just binary. Now we define the third part in the objective function:

$$\text{minimize} \left\{ \sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} \left| S_j - S_k \right| \right\} \quad (4)$$

This term (4) requires that whenever two paris j and k are connected in the matrix A, their sentiment scores S_j and S_k should be close.

Constraints for Opposite Sentiments: Along a similar line as the previous constraints, we define B, a $n \times n$ matrix, where $B_{jk} \in [0, 1]$ represents our confidence about pairs j and k having opposite sentiments. The value of B_{jk} where $a_j = a_k$ is based on whether w_j and w_k appear as antonyms in the thesaurus, and whether the pairs j and k are concatenated with conjunctives like "but" multiple times in the corpus.

However, the constraints of opposite sentiments are more complicated than those of similar sentiments, because we want their scores to be at the two extremes, so there is the sign of the sentiment score involved. Being opposite sentiment scores, the two scores are assumed to be in different signs (one positive and the other negative); at the same time, their absolute score values are assumed to be close.

In order to model this intuition, we separate the representation of *sign* and *absolute value* for each S_j by introducing two additional non-negative variables S_j^+ and S_j^-. We require S_j^+ and S_j^- both to be non-negative, but at most one of them is active (i.e. positive), the other being zero. In this way, (1) which variable being active represents the sign of S_j, i.e. S_j^+ being active is equivalent to S_j being positive; S_j^- being active is equivalent to S_j being negative; and (2) the value of the active variable (S_j^+ or S_j^-) represents the absolute value of S_j.

This idea of separating the representation of S_j's sign and absolute value is implemented as follows:

$$\text{minimize} \left\{ \sum_{j=1}^{n} (S_j^+ + S_j^-) \right\} \quad (5)$$

subject to

$$S_j = S_j^+ - S_j^- \qquad \text{for } j = 1 \cdots n \quad (6)$$

$$S_j^+, S_j^- \geq 0 \qquad \text{for } j = 1 \cdots n \quad (7)$$

Given the equality constraints on (6) (7), term (5) is essentially forcing at least one of S_j^+ and S_j^- to be zero. For example, if $S_j = 0.85$ and given no other constraints, the assignment of $S_j^+ = 0.85, S_j^- = 0$ will be favored over $S_j^+ = 1, S_j^- = 0.15$, as the first assignment minimizes $(S_j^+ + S_j^-)$.

Now that we can represent the sign and absolute value of each S_j separately, we define the fourth part of the objective function as follows:

$$\text{minimize} \left\{ \sum_{j=1}^{n} \sum_{k=1}^{n} B_{jk} \left(\left| S_j^+ - S_k^- \right| + \left| S_j^- - S_k^+ \right| \right) \right\} \quad (8)$$

Term (8) favors a solution in which if two instances S_j and S_k are connected in the opposite-sentiment matrix B, their sentiment signs are different but absolute values of sentiment scores are close.

5.3 Full Objective Function

Combining all the constraints defined above, we have the following full objective function :

$$\begin{aligned}
\mathbf{\Omega} &= \frac{\lambda_{prior}}{||I^G||_1} \sum_{j=1}^{n} I_j^G \left| S_j - G_j \right| && (9)\\
&+ \frac{\lambda_{rating}}{||I^O||_1} \sum_{i=1}^{m} I_i^O \left| \sum_{j=1}^{n} X_{ij} S_j - O_i \right| && (10)\\
&+ \frac{\lambda_{sim}}{||A||_1} \sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} \left| S_j - S_k \right| && (11)\\
&+ \frac{\lambda_{oppo}}{||B||_1} \sum_{j=1}^{n} \sum_{k=1}^{n} B_{jk} \left(\left| S_j^+ - S_k^- \right| + \left| S_j^- - S_k^+ \right| \right) && (12)\\
&+ \frac{\delta}{n} \sum_{j=1}^{n} (S_j^+ + S_j^-) && (13)
\end{aligned}$$

Now the optimization problem is

$$S = \text{argmin } \mathbf{\Omega} \quad (14)$$

subject to:

$$\begin{aligned}
S_j = S_j^+ - S_j^- &\qquad \text{for } j = 1 \cdots n\\
S_j^+, S_j^- \geq 0 &\qquad \text{for } j = 1 \cdots n\\
-1 \leq S_j \leq 1 &\qquad \text{for } j = 1 \cdots n
\end{aligned}$$

where $\lambda_{prior}, \lambda_{rating}, \lambda_{sim}, \lambda_{oppo}$ are weighting parameters which should be set to the degree that we trust each source of information, and δ can be set to a small value such as 0.01. For example, if we believe the similar-sentiment and opposite-sentiment information are of equal importance, we can set $\lambda_{sim} = \lambda_{oppo}$. The denominators in the form of $||M||_1$ represent the 1-norm of the corresponding vector or matrix M, i.e. the sum of all elements absolute values.

These are constants used to normalize the weighting parameters so that their impact is comparable. Note that, it is possible to use other loss functions in the objective function such as mean squared loss, but our specific choice can be transformed into efficient linear programming.

5.4 Transformation into Linear Programming

To solve the optimization problem efficiently, we can transform it into an equivalent linear programing problem. Basically, for each absolute-value term, we introduce one additional non-negative variable representing the non-negative absolute value. For example, we introduce $x_1, x_2, ..., x_n$ for the first part of objective function in (9) and replace $\sum_{j=1}^{n} I_j^G |S_j - G_j|$ with $\sum_{j=1}^{n} I_j^G x_j$ and two sets of additional constraints:

$$S_j - G_j \leq x_j \qquad \text{for } j = 1 \cdots n \text{ and } I_j^G = 1$$
$$-S_j + G_j \leq x_j \qquad \text{for } j = 1 \cdots n \text{ and } I_j^G = 1$$

The additional constraints imply that $x_1, x_2, ..., x_n$ are non-negative, so we do not need to explicitly list the non-negative constraints. Similarly, we can apply similar transformation to all the other terms in the objective function and obtain a linear programming problem where the objective function, equality and inequality constraints are all linear, i.e.

$$S = \text{argmin } \mathbf{\Omega} = \text{argmin}$$
$$\{ \frac{\lambda_{prior}}{||I^G||_1} \sum_{j=1}^{n} I_j^G x_j + \frac{\lambda_{rating}}{||I^O||_1} \sum_{i=1}^{m} I_i^O y_i + \frac{\lambda_{sim}}{||A||_1} \sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} z_{ij}$$
$$+ \frac{\lambda_{oppo}}{||B||_1} \sum_{j=1}^{n} \sum_{k=1}^{n} B_{jk}(u_{jk} + u_{kj}) + \frac{\delta}{n} \sum_{j=1}^{n} (S_j^+ + S_j^-) \}$$

subject to

$$S_j = S_j^+ - S_j^- \qquad \text{for } j = 1 \cdots n$$
$$S_j^+, S_j^- \geq 0 \qquad \text{for } j = 1 \cdots n$$
$$-1 \leq S_j \leq 1 \qquad \text{for } j = 1 \cdots n$$
$$S_j - G_j \leq x_j \qquad \text{for } j = 1 \cdots n \text{ and } I_j^G = 1$$
$$-S_j + G_j \leq x_j \qquad \text{for } j = 1 \cdots n \text{ and } I_j^G = 1$$
$$\sum_{j=1}^{n} X_{ij} S_j - O_i \leq y_i \qquad \text{for } i = 1 \cdots m \text{ and } I_j^O = 1$$
$$-\sum_{j=1}^{n} X_{ij} S_j + O_i \leq y_i \qquad \text{for } i = 1 \cdots m \text{ and } I_j^O = 1$$
$$S_j - S_k \leq z_{jk} \qquad \text{for } j, k = 1 \cdots n \text{ and } A_{j,k} > 0$$
$$-S_j + S_k \leq z_{jk} \qquad \text{for } j, k = 1 \cdots n \text{ and } A_{j,k} > 0$$
$$S_j^+ - S_k^- \leq u_{jk} \qquad \text{for } j, k = 1 \cdots n \text{ and } B_{j,k} > 0$$
$$-S_j^+ + S_k^- \leq u_{jk} \qquad \text{for } j, k = 1 \cdots n \text{ and } B_{j,k} > 0$$

An important and nice theoretic property of linear programming is that the linear constraints define the feasible region, which is a convex polyhedron; and a linear objective function is also a convex function, which implies that every local minimum is a global minimum. By transforming our optimization problem into an equivalent linear programming problem, we can utilize many known methods and toolkits to solve it efficiently. Since the construction of sentiment lexicon is an offline task, no real-time response is required. But still, all the experiments on our data sets finished within a few seconds.

	Hotel Data	Printer Data
Domain Specific Sentiments	ROOM:private + FOOD:excelent + LOCATION:farthest - FOOD:tiny -	SOFTWARE:compatible + QUALITY:professional + ERRMSG:frequently - SUPPORT:eventually -
Aspect Dependent Sentiments	ACTIVITIES:inside - FACILITIES:inside + ROOM:huge + PRICE:huge - ACTIVITIES:cool + SERVICE:cool -	QUALITY:high + NOISE:high - INK:cheap + APPEARANCE:cheap - INK:fast + SUPPORT:fast -

Table 1: Sample Results of OPT

6. EXPERIMENTS

In this section, we present the experimental evaluation of our techniques. Our experiments employ two data sets from very different domains: one is hotel reviews from TripAdvisor (hotel data); the other is customer feedback survey for printers (printer data). Following most previous works, we extract adjectives and adverbs as candidate opinion words, although our method is general enough to score candidate opinion words in any part-of-speech. A WordNet-based lemmatizer is employed to transform each word to its original form (e.g. "checked" to "check"). For solving the linear programming problem, we use GAMS/CPLEX, which solves our problems within a few seconds on a machine with 2.80 GHz CPU and 2GB memory. The default setting used in the proposed optimization framework (**OPT**) is $\lambda_{prior} = \lambda_{sim} = \lambda_{oppo} = \lambda_{rating}$.

As comparison, we also consider the following baselines for learning a context-dependent sentiment lexicon:

- **Random**: for each aspect-opinion pair, simply predict its sentiment by random guessing, i.e. 33.33% as positive (+1), 33.33% as negative (-1), and 33.33% as neutral (0).
- **MPQA**: for each aspect-opinion pair j, simply predict its sentiment by looking at the sentiment of the opinion word w_j in the general-purpose sentiment lexicon MPQA[5].
- **INQ**: same as the previous method, except that General Inquirer[6] is used instead of MPQA.
- **Global**: the Global Prediction method proposed in [12]. It uses only the overall ratings to generate a context-dependent sentiment lexicon with a Naive Bayes method.

Note that, we are aware of two other methods in addition to the Global method that can output aspect-dependent sentiment scores. But the idea in [1] is similar to the Global method; and the other method [20] has a strict requirement that each text should come with all k aspects, which is not realistic and does not hold in our data sets. Thus, we only include the Global method here as a representative of state-of-the-art.

6.1 Sample Results

We first present some interesting sample results in the context-dependent sentiment lexicon constructed by our optimization framework. From Table 1, we can see that

1. Our method picked up domain-specific new sentiment words that are not in any general-purpose sentiment lexicon. For example, "private" is positive in the ho-

[5] http://www.cs.pitt.edu/mpqa/
[6] http://www.wjh.harvard.edu/~inquirer/

tel domain and "compatible" is positive in the printer domain. In addition, our method can detect correct sentiment even when the spelling is wrong, e.g. "excelent". That is because we consolidate different statistical evidences to infer its meaning rather just looking at the matching string in the general lexicon.

2. Even in the same domain, our method also identified different sentiments for the same word depending on the aspects. For example, in hotel reviews: "huge room" conveys positive sentiment while "huge price" is not desirable. It is negative if the activities are "inside", but it is positive if the facilities are "inside" rather than "outside". Similarly, in the printer data, "high quality" is good but "high noise" is bad. People are happy if the ink is "cheap", but they are not happy about the "cheap appearance". The word "fast" has a negative connotation for "ink" (e.g. "ink runs out fast"), but it is positive if the support service is "fast".

6.2 Evaluation of Lexicon Quality

There is no existing data set available to evaluate the quality of a constructed context-dependent sentiment lexicon, which is in the form of a sentiment score assigned to each aspect-opinion pair. In this section, we describe how we create a gold standard by performing human annotation on a data set of hotel reviews from TripAdvisor. By comparing against this gold standard, we evaluate the lexicons constructed using different methods.

6.2.1 Hotel Data

Data Description: We collected 4792 reviews about a well-known hotel brand from TripAdvisor. Each review has an overall rating (between 1 and 5 stars) of the hotel from the user in addition to the review text. We manually specified 7 aspects in the hotel domain, i.e., Location, Food, Room, Facilities, Service, Value and Activities. For example, the aspect or word cluster "LOCATION" contains words like: downtown, shuttle, metro, airport and etc.

Human Annotation: We randomly sample 750 reviews out of 4792 reviews to be labeled by 5 human judges, and each review is ensured to be labeled by 2 judges. For each sentence with extracted candidate aspect-opinion pairs (using the method described in Section 5.1), we display the original sentence to the judges followed by the tuples in the format of "aspect:attribute:opinion". The judges are asked to label each tuple with one of the following tags:

+: if positive in the context

-: if negative in the context

0: if neutral in the context

N: if do not apply

X: if attribute-aspect mapping is wrong

Below we show an instance that the judge will see.

```
"within 10 mins , we were checked in and on
our way to our room , which was fantastic."
SERVICE:check_in:fantastic
ROOM:room:fantastic
```

Note that, there may be ambiguities. In the above example, judges may have their own opinions about whether "fantastic" applies to "SERVICE" or "ROOM"or both. Considering all occurrences of aspect-opinion pairs which are labeled with +, -, or 0, the average agreement among human annotators is 78.18% which is comparable to what had been reported in existing work of sentiment analysis [18].

Method	Precision	Recall	F-Measure
Random	0.4932	0.2784	0.3559
MPQA	**0.9631**	0.3702	0.5348
INQ	0.8757	0.4397	0.5855
Global	0.7073	0.5929	0.6451
OPT	0.8125	**0.6823**	**0.7417**

Table 2: Lexicon Quality Evaluation on Hotel Data

Gold Standard: After collecting the labels from human judges, we filter aspect-opinion pair occurrences to keep only the 3730 occurrences agreed by both judges. Then we aggregate those instances into 1127 unique pairs. To alleviate the ambiguity problem, we create our gold standard sentiment lexicon by using only the 705 aspect-opinion pairs labeled +1 or -1, which tend to represent high confidence and consistency of the labels. This gold standard lexicon is domain specific and aspect-dependent as well; it contains high-quality entries agreed by human annotators. But the coverage is relatively small because we only include the high-confident ones in the gold standard in order to be accurate.

6.2.2 Evaluation Measures

Since the gold standard sentiment lexicon contains only binary labels (either +1 or -1), we first transform our output sentiment lexicon into the same format by only considering the sign of the predicted sentiment value, so that the assigned scores are either +1 or -1. After that, the output sentiment lexicon can be evaluated by:

$$\textbf{precision} = \frac{N_{agree}}{N_{lexicon}}$$

$$\textbf{recall} = \frac{N_{agree}}{N_{gold}}$$

$$\textbf{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where N_{gold} is the number of aspect-opinion pairs in the gold standard lexicon, $N_{lexicon}$ is the number of aspect-opinion pairs in the automatically constructed sentiment lexicon (i.e. 705), N_{agree} is the number of pairs that are consistently labeled (either both +1 or both -1) in the gold standard and constructed lexicons.

6.2.3 Results

Note that the human annotation is for evaluation purpose only, and the automatic algorithms do not use any labels. So we run the algorithms on the whole set of 4792 reviews instead of the subset of 750 reviews labeled by human judges. After generating candidate lexicon entries, we extract 4627 unique aspect-opinion pairs with at least two occurrences, and score them with different algorithms. However, as there are only 705 pairs in the gold standard, there is some bias in the evaluation by precision. This is because that there can be some aspect-opinion pairs correctly output by the algorithms but they do not appear in the subset of 750 reviews so human annotators did not label them. As a result, the precision should be taken with a grain of salt here. Take an extreme example: a naive method outputting only

one correct pair (e.g. "LOCATION:excellent:+1") will have 100% precision but extremely low recall; but it is not useful in practice. Thus, F-measure should be a more reliably measure in order to evaluate the usefulness of a sentiment lexicon, because it captures the balance between precision and recall.

The results of different methods on hotel data are shown in Table 2 where the best performance under each measure is highlighted in bold font. We can see that when directly evaluating the lexicon quality,

- Dictionary-based baselines (i.e. MPQA and INQ) which totally ignore the context, provide best precision performance, at the price of low recall. The recall of MPQA and INQ is significantly lower than other methods that take context into consideration (Global and OPT). This suggests that there are a lot of domain specific and aspect dependent words that carry sentiments but are totally ignored by dictionary-based baselines.
- In comparison, the Global method, gives a better balance of precision and recall and thus better F-measure. This method is able to pick up domain specific and context dependent sentiments by exploiting the association among aspects, words and document-level overall rating.
- Our method OPT further improves the Global method in *both precision and recall* significantly (and thus F-measure too, by almost 15%). This is because that in addition to the overall rating OPT also incorporates the prior sentiments from dictionaries, the similar/opposite sentiment information and linguistic heuristics, which help the sentiment prediction especially when the signal from the overall ratings is not present or not strong enough to tell the sentiments of some words.

6.3 Evaluation of Aspect-Level Sentiment Classification Using the Lexicon

The value of a sentiment lexicon mostly lies in its use in applications. Thus, in addition to the evaluation of the lexicon quality, we also conduct experiments to evaluate aspect-level sentiment classification performance of using different lexicons. The task is to produce a sentiment score for a given aspect in a piece of text, e.g. whether a particular hotel review is talking positively or negatively about the LOCATION aspect.

6.3.1 Hotel Data

From the manually annotated hotel data described in Section 6.2.1, we use the sentiments at each review-aspect level as the gold standard. Again, in order to ensure the high confidence of the gold standard, we only consider those aspect-opinion pairs that have labels agreed by two judges. After that, the gold standard sentiment at each review-aspect level is the averaged sentiment labels of the corresponding aspect-opinion pairs in the review, which is a real value between -1 and $+1$.

6.3.2 Printer Data

Data Description: For the second data set, we obtain 3511 customer feedback surveys about a printer brand. Each survey comes with an overall satisfaction rating (between 1 and 5) and a small piece of text of detailed comments (usually just one or two sentences).

Statistics	Hotel Data	Printer Data
# of reviews	750	3511
# of possible aspects	7	25
AVG # of aspects per review	2.86	1.32
AVG # words per review	270	24

Table 3: Data Set Statistics for Sentiment Classification Task

Human Annotation: The company manufacturing the printers hired people to manually label the feedback text so as to get deeper understanding about what people are happy about their printers and what they are upset about. The human judges are provided with an aspect description file, in which a set of aspect tags are defined by a short description. For example

```
[TRIES]: The number of unsuccessful tries
         before install success.
[INK]: Ink and print head related issues
       (Including Install and Removal).
```

During the labeling process, the judges read each survey, tag it with the matching aspect tags, and assign a sentiment score among $\{-3, -2, -1, +1, +2, +3\}$ for each aspect tag. For instance, the review text of "Easy to set up. digital monitoring is great for ink needs. " is tagged as "[+3, TRIES] " and "[+3, INK]", because it is talking very positively about both the "TRIES" and the "INK" aspects. Then we use the top 25 most frequently tagged aspects in our experiments. Unfortunately, we do not know further details such as how many human judges are involved and what is their agreement, so we cannot report them here.

Both the hotel data and the printer data are manually labeled with different sentiment scores for each document-aspect combination. This enables us to evaluate the aspect-level sentiment classification performance of using different sentiment lexicons, which represents a real application need. Actually the classification results are essentially what the printer company is interested in. If we can do accurate classification automatically, we can save companies effort to hire people to label the aspect-level sentiment. Some statistics about the two data sets are summarized in Table 3.

6.3.3 Evaluation Scheme and Measures

For the task of sentiment classification at the document-aspect level, we need to first use a sentiment lexicon to predict the sentiment score for each document-aspect combination. Since we only use an unlabeled corpus, we will continue using unsupervised method for the prediction. In particular, we adopt the following simple but reasonable baseline approach: for each document-aspect combination (d_i, a_j),we identify all the aspect-opinion pairs on the aspect a_j occurring in document d_i, look up the sentiment score of each pair in the context-dependent sentiment lexicon, and then take the average of sentiment scores as the predicted score for this combination (d_i, a_j).

Now if we only consider the binary sign of the sentiment scores, we can also use precision, recall, and F-measure for evaluation. But as the gold standard scores are real values (all normalized to $[-1, 1]$ by min-max normalization) rather than being binary, we also include **Mean Squared Error (MSE)** as an additional measure, which measures the distance between the predicted sentiment and the gold

Method	Prec	Recall	F-Measure	MSE
HOTEL DATA				
Random	0.4368	0.3689	0.3999	0.567
MPQA	**0.8128**	0.5289	0.6408	0.47
INQ	0.78	0.6294	0.6966	0.4561
Global	0.6975	0.773	0.7333	0.4426
OPT	0.7283	**0.7756**	**0.7512**	**0.416**
PRINTER DATA				
Random	0.4844	0.2629	0.3408	0.7142
MPQA	0.7579	0.1597	0.2639	0.574
INQ	0.7879	0.3502	0.4849	0.5365
Global	0.7645	**0.5448**	0.6362	0.5091
OPT	**0.8222**	0.5276	**0.6428**	**0.468**

Table 4: Sentiment Classification Performance

standard sentiment. MSE is more an accurate measure in the sense that it captures the notion that classifying a positive class into a negative class is worse than classifying it into a neutral one. Lower MSE means better classification accuracy.

6.3.4 Results

We summarize the results on both data sets in Table 4 and highlighted in bold font the best performance under each measure.

In the aspect-level sentiment classification task, which is a real application of the constructed context-dependent sentiment lexicon,

- dictionary-based baselines (MPQA and INQ) do not necessarily gives best precision. Moreover, they suffer more at recall on the printer data. (Especially, recall of MPQA is even lower than the random baseline.)
- The Global method still performs well on both precision and recall.
- Our OPT method provides the best balance between precision and recall; it achieves the best F-measure performance on both data sets.
- Furthermore, when we zoom into the performance evaluated at finer granularity, i.e. as measured by MSE, the performance gain of OPT is even more significant. It has reduced the best MSE in the baselines from 0.4426 to 0.416, from 0.5091 to 0.468 on the two data sets respectively, both improvements are statistically significant with p-value less than 10^{-6} in a paired t-test.

All these observations suggest that a lexicon with higher precision (as shown by dictionary-based baselines in Table 2 where we directly evaluate the lexicon quality) does not necessarily lead to better aspect-level classification performance. The low recall of the dictionary-based baselines would result in many misses of domain-specific and aspect-dependent polarity words, thus lead to less accurate classification of aspect-level sentiment. Thus, it is important to achieve a good balance between precision and recall. In particular, if one is mainly interested in aspect-level classification, which is one of the most important applications of sentiment lexicons, OPT is by far the best method. Such performance advantage demonstrates the effectiveness of combining multiple useful signals in our optimization framework.

	λ_{prior}	λ_{rating}	λ_{sim}	λ_{oppo}	F-Measure
Default	1	1	1	1	0.7417
Drop	0	1	1	1	0.6549
one	1	0	1	1	0.6453
term	1	1	0	1	0.7309
	1	1	1	0	0.7408
Weighting	2	2	1	1	0.7431
important	3	3	1	1	**0.7544**
terms	6	6	1	1	0.7510
	8	8	1	1	0.7506

Table 5: OPT Parameter Tuning: Lexicon Quality on Hotel Data

6.4 Analysis of Parameter Tuning

We have already shown that OPT in the default parameter setting outperforms all baselines on both lexicon quality evaluation and sentiment classification evaluation. Now we further look into the four parameters $\lambda_{prior}, \lambda_{sim}, \lambda_{oppo}, \lambda_{rating}$ that basically weight the importance of the four components in the objective function. Our framework is very general, and if we set one parameter to zero it is equivalent to not using the signal as defined in the corresponding term. For the purpose of examining the importance of different signals, we conduct some analysis experiments where one term is dropped out in each experiment.

Lexicon Quality: The middle rows in Table 5 show the lexicon quality evaluation results of "dropping one term" tested on the hotel data. Due to the space limit, we only display the F-measure here. It can be seen that (1) dropping any term in the objective function decreases the lexicon quality, indicating that all the constraints are useful. (2) when setting λ_{prior} or λ_{rating} to zero, the performance decreases dramatically (F-measure from 0.7417 to around 0.65), which suggests that these two terms contain more important information. Then we tried to place more weights on the two important terms. As shown in the bottom four rows, performance can be further increased, where the best one is highlighted in bold font.

Classification Performance: In Table 6, we also show results of parameter tuning on the sentiment classification task. Similar trend is observed too, i.e. classification performance is improved if we put more weights on the important signals. One thing to note is that the importance of signals is different in the two data sets: both the prior sentiments and the overall ratings are important in the hotel data while the overall ratings serve as the most important signal in printer data.

This series of experiments demonstrate that our optimization framework is general enough to accommodate different weights placed on different kinds of signals for constructing a context-dependent sentiment lexicon, which can lead to even better performance than the default setting. This is especially useful when we have some reliable prior belief of the importance of signals; then we can put more weights on more important signals. Nevertheless, there is still the challenge of automatically setting the optimal parameters for different domains and/or different data sets, which we intend to study as future work.

					Hotel	Data					Printer	Data
	λ_{prior}	λ_{rating}	λ_{sim}	λ_{oppo}	**F-Measure**	**MSE**	λ_{prior}	λ_{rating}	λ_{sim}	λ_{oppo}	**F-Measure**	**MSE**
Default	1	1	1	1	0.7512	0.416	1	1	1	1	0.643	0.468
Drop	0	1	1	1	0.7396	0.4436	0	1	1	1	0.656	0.467
one	1	0	1	1	0.6629	0.4749	1	0	1	1	0.453	0.673
term	1	1	0	1	0.7733	0.4057	1	1	0	1	0.657	0.446
	1	1	1	0	0.7508	0.4132	1	1	1	0	0.649	0.468
Weighting	2	2	1	1	0.7632	0.4096	1	2	1	1	0.662	0.459
important	3	3	1	1	0.7737	0.4054	1	3	1	1	0.668	0.456
terms	6	6	1	1	0.7781	0.4015	1	6	1	1	0.671	0.451
	8	8	1	1	**0.7794**	**0.4008**	1	8	1	1	**0.672**	**0.449**

Table 6: OPT Parameter Tuning: Sentiment Classification Performance on Both Data Sets

7. CONCLUSION AND FUTURE WORK

In this paper we studied the problem of automatically constructing a context-dependent sentiment lexicon from an unlabeled opinionated text collection. We studied and summarized several kinds of useful signals, formulated an optimization problem to combine all the signals, and provided a mathematical transformation into linear programming. We have demonstrated that our method can learn new domain specific sentiment words and aspect-dependent sentiment. Further quantitative evaluation against baselines and a state-of-the-art method shows that (1) for a given domain our framework can greatly improve the coverage of a general sentiment lexicon; (2) constructed aspect-level sentiment lexicons are in good quality, achieving a good balance of precision and recall; (3) sentiment classification performance can be significantly improved with the automatically constructed context-dependent sentiment lexicon; and (4) parameter tuning gives more performance advantage.

The framework we proposed is quite general and applicable for opinionated text collection in any domain. It is capable of incorporating different sources of available information for the automatic construction of a context-aware sentiment lexicon. As future work, we can exploit other kinds of useful signals such as "pros" and "cons" sections in the reviews and aspect-level ratings. We also plan to evaluate the effectiveness of our context-aware sentiment lexicon in other sentiment related applications, such as opinion retrieval and opinion summarization. Another interesting future work is to study how to tune the weighting parameters automatically for optimal performance.

8. REFERENCES

[1] K. T. Chan and I. King. Let's tango — finding the right couple for feature-opinion association in sentiment analysis. In *PAKDD '09*, pages 741–748, 2009.

[2] Y. Choi and C. Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *EMNLP '09*, pages 590–598, 2009.

[3] H. T. Dang. Overview of the tac 2008 opinion question answering and summarization tasks. In *TAC*, 2008.

[4] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM '08*, pages 231–240.

[5] A. Hassan and D. Radev. Identifying text polarity using random walks. In *ACL '10*, pages 395–403.

[6] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *EACL '97*, pages 174–181, 1997.

[7] V. Hatzivassiloglou and J. M. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING '00*, pages 299–305, 2000.

[8] L. Hoang, J.-T. Lee, Y.-I. Song, and H.-C. Rim. Combining local and global resources for constructing an error-minimized opinion word dictionary. In *PRICAI '08*, pages 688–697, Berlin, Heidelberg, 2008. Springer-Verlag.

[9] V. Jijkoun, M. de Rijke, and W. Weerkamp. Generating focused topic-specific sentiment lexicons. In *ACL '10*, pages 585–594, 2010.

[10] Y. Jo and A. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM '11*.

[11] H. Kanayama and T. Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP '06*, pages 355–363, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[12] Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *18th International World Wide Web Conference (WWW2009)*, April 2009.

[13] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07*, pages 171–180. ACM.

[14] S. Mohammad, C. Dunne, and B. Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *EMNLP '09*, pages 599–608, 2009.

[15] S.-H. Na, Y. Lee, S.-H. Nam, and J.-H. Lee. Improving opinion retrieval based on query-specific sentiment lexicon. In *ECIR '09*, pages 734–738, Berlin, Heidelberg, 2009. Springer-Verlag.

[16] A. Neviarouskaya, H. Prendinger, and M. Ishizuka. Sentiful: Generating a reliable lexicon for sentiment analysis. In *ACII*, pages 1 –6, sep. 2009.

[17] I. Ounis, M. D. Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the trec 2006 blog track. In *TREC. NIST*, 2006.

[18] B. Pang and L. Lee. *Opinion Mining and Sentiment Analysis*, volume 2(1–2) of *Foundations and Trends in Information Retrieval*. Now Publ., 2008.

[19] P. D. Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, 2003.

[20] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD '10*, pages 783–792, New York, NY, USA, 2010. ACM.

[21] J. Yi, T. Nasukawa, R. C. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of ICDM 2003*, pages 427–434, 2003.

[22] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP '03*, pages 129–136, 2003.