Propagation-based Sentiment Analysis for Microblogging Data

Jiliang Tang*, Chikashi Nobata†, Anlei Dong†, Yi Chang† and Huan Liu*

Abstract

The explosive popularity of microblogging services encourages more and more online users to share their opinions, and sentiment analysis on such opinion-rich resources has been proven to be an effective way to understand public opinions. On the one hand, the brevity and informality of microblogging data plus its wide variety and rapid evolution of language in microblogging pose new challenges to the vast majority of existing methods. On the other hand, microblogging texts contain various types of emotional signals strongly associated with their sentiment polarity, which brings about new opportunities for sentiment analysis. In this paper, we investigate propagation-based sentiment analysis for microblogging data. In particular, we provide a propagating process to incorporate various types of emotional signals in microblogging data into a coherent model, and propose a novel sentiment analysis framework PSA which learns from both labeled and unlabeled data by iteratively alternating a propagating process and a fitting process. We conduct experiments on real-world microblogging datasets, and the results demonstrate the effectiveness of the proposed framework. Further experiments are conducted to probe the working of the key components of the proposed framework.

1 Introduction

Nowadays microblogging services such as Twitter, Tumblr [6] and Chinese microblogging website Weibo are increasingly used by online users to share and exchange opinions, providing rich resources to understand public opinions. For example, in [3], a simple model exploiting Twitter sentiment and content outperforms market-based predictors in terms of forecasting box-office revenues for movies; public mood as measured from a large-scale collection of tweets obtains an accuracy of 86.7% in predicting the daily up and down changes in the closing values of the DJIA [5]. Therefore sentiment analysis for such opinion-rich resources has attracted increasing attention in recent years [8, 26, 31, 12].

Two categories of methods are extensively studied for sentiment analysis: supervised methods and unsupervised methods [23, 15]. Supervised methods build a classifier trained on manually labeled data [24], while unsupervised methods, such as the lexicon-based methods, determine the sentiment of texts based on a sentiment lexicon [30, 20]. These methods have been successfully applied to various domains such as product and movie reviews [24, 10]. However, both of them become difficult when handling microblogging data. First, informal and abbreviated language is widely used in microblogging and rapidly evolves. To adapt to changes in language use, supervised methods need new labeled training data, while unsupervised lexicon-based methods also require human efforts to update the sentiment lexicons. Both are time and labor consuming. Second, microblogging texts are short and often do not provide sufficient statistical information. Due to the large feature space and the brevity of microblogging data, supervised methods need more labeled data [16], while short microblogging texts lack sufficient aggregation information to measure their overall sentiment for the unsupervised lexicon-based methods [11]. The combination of these two distinct characteristics of microblogging data manifests a new research challenge for sentiment analysis and calls for adaptive methods with less manual efforts.

It is common to have a microblogging dataset with a small number of labeled data and a large amount of unlabeled data. In addition to providing context information for labeled data such as changes in language use [26], unlabeled data in microblogging contains various types of emotional signals. Microblogging texts are not created independently and there exist sentiment correlations, which can be explained by sentiment consistency [1] and emotional contagion [9]. In the physical world, people often use gestures and facial expressions to indicate their emotions. Similarly, in microblogging, users develop visual cues such as emoticons that are strongly associated with their emotional states. When users adopt emoticons, they are effectively associating the text with an emotional state [17]. The availability of unlabeled data with emotional signals brings about new opportunities for sentiment analysis and enables the development of frameworks by exploiting both labeled and unlabeled data.

These unique properties of microblogging data motivate the development of a propagation-based sentiment analysis framework. In essence, we investigate - (1) how to exploit various types of emotional signals in microblogging data; and (2) how to take advantage of labeled and unlabeled data with emotional signals for sentiment analysis. Providing solutions to these two questions results in a novel propagation-based sentiment analysis framework PSA. The proposed framework iteratively alternates two processes - a propagating process and a fitting process. Our main contributions are summarized below:

- Provide a propagating process to incorporate various types of emotional signals into a coherent model;
- Propose a propagation-based sentiment analysis framework which makes use of both labeled and unlabeled

*Computer Science and Engineering, Arizona State University, Tempe, AZ. {jiliang.tang, xia.hu, huiji.gao, huan.liu}@asu.edu; † Yahoo! Labs, Sunnyvale, CA 94089.{chikashi, anlei, yichang}@yahoo-inc.com

data with emotional signals by iteratively alternating a propagating process and a fitting process;

- Evaluate the proposed framework PSA extensively in two real-world microblogging datasets to understand the working of PSA.

2 The Proposed Semi-supervised Sentiment Analysis Framework

Before going into the details of our framework, we first introduce some important notations used in this paper. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ and $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ be the microblog set and the dictionary respectively, where N is the number of microblogs and m is the size of the dictionary. The matrix representation of $\mathcal{T}$ is $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{m \times N}$. Assume that there are n microblogs in $\mathbf{X}$ labeled. Let $\mathbf{X} = [\mathbf{X}^L, \mathbf{X}^U]$ where $\mathbf{X}^L \in \mathbb{R}^{m \times n}$ is the labeled set and $\mathbf{X}^U \in \mathbb{R}^{m \times (N-n)}$ is the unlabeled set. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$ be the set of sentiment classes where c is the number of classes. There are two common choices for c in the literature including $c = 2$, i.e., $\{C_1 = negative,\ C_2 = positive\}$, and $c = 3$, i.e., $\{C_1 = negative,\ C_2 = neutral,\ C_3 = positive\}$. $\mathbf{Y} \in \mathbb{R}^{c \times n}$ is the label indicator matrix for labeled data where $\mathbf{Y}_{ji} = 1$ if the i-th microblog belongs to the j-th class, and 0 otherwise.

Microblogs are not created in isolation [26, 12]. We can exploit social theories such as sentiment consistency [1] and emotional contagion [9] to construct a microblog-microblog sentiment correlation network. For example, via hypothesis testing, [12] demonstrates that the sentiments of two microblogs created by the same user in a short time window are likely to be similar. We construct a microblog-microblog network $\mathbf{R}^{tt}$ based on these social theories where $\mathbf{R}^{tt}_{ij} = 1$ if the sentiments of t_i and t_j may be correlated, zero otherwise. Similarly, words in microblogging texts are also not independent and their sentiment polarity may be correlated. For example, two words appearing frequently in the same set of microblogs are likely to share similar sentiment polarity [11]. We construct a word-word network $\mathbf{R}^{ww}$ where $\mathbf{R}^{ww}_{ij} = 1$ if d_i and d_j may be correlated, zero otherwise. More details about how to construct these two networks will be further discussed in the experiment section. We use $\mathbf{R}^{tw}$ to denote the microblog-word bipartite graph where $\mathbf{R}^{tw}_{ij}$ denotes the frequency of d_j in t_i. Microblogs, words and their relations are demonstrated in the top subgraph of Figure 1 where microblogs may contain emoticons, and some words' sentiment polarity can be indicated by existing lexicons.

In this paper, we propose a novel propagation-based sentiment analysis framework PSA for microblogging data, which is demonstrated in Figure 1. The framework includes two processes - a propagating process and a fitting process. The propagating process propagates the classification results and emotional signals such as emoticons via word-word, microblog-word and microblog-microblog relations as shown in the top subgraph of Figure 1; while the fitting process learns a classifier to fit both the label information and propagation results as shown in the bottom subgraph of Figure 1. We will iteratively alternate the propagating process and the fitting process until convergence. More details about these two processes will be discussed in the following subsections.

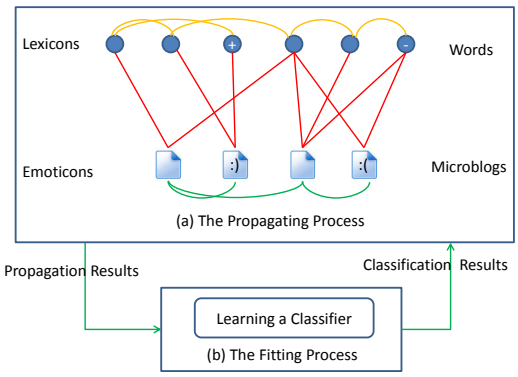


Figure 1: The Proposed Framework.

2.1 The Propagating Process

A microblog is likely to be positive if it contains many words with positive sentiment, and connects to other microblogs with positive sentiment; meanwhile a word is likely to be positive if it associates with many microblogs with positive sentiment, and correlates to other words with positive sentiment, which can also be applied to other sentiment classes such as negative and neutral. These observations reveal the mutual reinforcement principle among sentiments of microblogs and words, which paves a way for us to model the propagating process.

Let $\mathbf{p}_i \in \mathbb{R}^c$ and $\mathbf{q}_j \in \mathbb{R}^c$ denote the sentiment of the i-th microblog t_i and the j-th word d_j, respectively. With the mutual reinforcement principle, the propagating process can be formulated as,

$$\begin{aligned} \mathbf{p}_i &= \alpha \frac{\sum_{k=1}^{N} \mathbf{R}^{tt}_{ik} \mathbf{p}_k}{\sum_{k=1}^{N} \mathbf{R}^{tt}_{ik}} + (1-\alpha) \frac{\sum_{k=1}^{m} \mathbf{R}^{tw}_{ik} \mathbf{q}_j}{\sum_{k=1}^{m} \mathbf{R}^{tw}_{ik}}, \\ \mathbf{q}_j &= \beta \frac{\sum_{k=1}^{m} \mathbf{R}^{ww}_{jk} \mathbf{q}_k}{\sum_{k=1}^{m} \mathbf{R}^{ww}_{jk}} + (1-\beta) \frac{\sum_{k=1}^{N} \mathbf{R}^{tw}_{jk} \mathbf{p}_k}{\sum_{k=1}^{N} \mathbf{R}^{tw}_{kj}}, \end{aligned} \tag{2.1}$$

where the sentiment of a microblog (or a word) is an aggregation of the sentiment of its correlated microblogs (or presented microblogs) and contained words (or correlated words). Via Eq. (2.1), the sentiment is propagated in word-word, microblog-word and microblog-microblog relations. That is why we call this process the propagating process. α and β are two parameters to control the contributions from microblog-microblog relations and word-word relations respectively. We empirically find that $\alpha = \beta = 0.4$ works well in this paper. The initialization of the propagation process will be presented in Algorithm 1.

One important issue about the propagation process is how to initialize $\mathbf{p}_i$ and $\mathbf{q}_j$. For t_is and d_js, we usually will use the classification results and the learnt classifier from the fitting process to perform initialization, which will be discussed in the following subsections. However, there are other types of emotional signals such as emoticons and existing sentiment lexicons available for some microblogs and words. Next we will discuss how to incorporate these types of emotional signals in the propagation process.

Incorporating Emoticons : Users in microblogging usually use emoticons to indicate their sentiment, and emoticons in microblogs are strong indicators of the sentiment of microblogs [11]. We use $\mathcal{E}_k$ to denote the set of microblogs from unlabeled data with emoticons of C_k, and $\mathbf{p}_i$ will be set as $(\overbrace{0,\ldots,0}^{k-1},1,\overbrace{0,\ldots,0}^{c-k})$. For example, assume that $c=2$ and we use $\mathcal{E}_1$ and $\mathcal{E}_2$ to denote sets of microblogs with negative and positive emoticons from unlabeled data, respectively. Since tweets with positive emoticons (or negative emoticons) are very likely to be positive (or negative) [11], for the i-th microblog t_i with emoticon information, we initialize the sentiment $\mathbf{p}_i$ as

$$\mathbf{p}_i = \begin{cases} (1,0), & \text{if } t_i \in \mathcal{E}_1 \\ (0,1), & \text{if } t_i \in \mathcal{E}_2 \end{cases}. \tag{2.2}$$

To capture emoticon information in t_i, we fix $\mathbf{p}_i$ during the propagating process instead of being updated by Eq. (2.1).

Incorporating prior knowledge from lexicons : There are existing lexicons to indicate the sentiment polarity of words that can be used as prior knowledge to improve sentiment analysis performance. We use $\mathcal{E}_k$ to denote the set of words with the sentiment of C_k indicated by the sentiment lexicons, and $\mathbf{q}_j$ will be set as $(\overbrace{0,\ldots,0}^{k-1},1,\overbrace{0,\ldots,0}^{c-k})$. Similar to incorporating emoticon information, we fix the sentiment of the j-th word $\mathbf{q}_j$ during the propagating process if d_j is in the lexicons.

The significance of the propagating process is two-fold. First, it provides a unified way to incorporate various types of emotional signals such as sentiment correlations, emoticons and lexicons. Second, via the propagating process, sentiment from emotional signals and label information in the fitting process can be propagated to unlabeled data, which allows us to learn from unlabeled and labeled data. Next we will introduce the fitting process.

2.2 The Fitting Process In the fitting process, our framework will learn a classifier to fit both label information and propagation results from the propagating process. In this paper, we assume that there is a linear classifier to classify the sentiment of microblogs $\mathbf{W} \in \mathbb{R}^{m\times c}$ and the fitting process can be formulated as,

$$\begin{aligned}\min_{\mathbf{W}\geq 0}\ F(\mathbf{W}) &= l(\mathbf{W}^\top \mathbf{X}^L, \mathbf{Y}) + \lambda_T \sum_{i=1}^{N} d(\mathbf{p}_i, \mathbf{W}^\top \mathbf{x}_i) \\ &+ \lambda_W \sum_{j=1}^{m} d(\mathbf{q}_j, \mathbf{W}_j) + \lambda \|\mathbf{W}\|_1,\end{aligned} \tag{2.3}$$

where $l(\cdot,\cdot)$ is a loss function and popular choices include square loss, logistic loss and hinge loss. $d(\cdot,\cdot)$ is a distance metric. $\mathbf{W}_j$ is the j-th row of $\mathbf{W}$. Next we give more details about Eq. (2.3).

In Eq. (2.3), the first term is to fit the label information $\mathbf{Y}$. The second term is to fit the propagation results for microblogs from the propagation process. Since the second term is to fit the propagation of microblogs, we name it as *tPropagation* and λ_T is introduced to control its contribution. To understand tPropagation better, we divide the set of microblogs $\mathcal{T}$ into three subsets, $\mathcal{T} = \{\mathcal{T}^L, \mathcal{T}^E, \mathcal{T}^U\}$ where $\mathcal{T}^L$ and $\mathcal{T}^E$ denote microblogs with labels and emoticons respectively, and $\mathcal{T}^U$ is the remaining microblogs. Then tPropagation can be rewritten as

$$\begin{aligned}\sum_{i=1}^{N} d(\mathbf{p}_i, \mathbf{W}^\top \mathbf{x}_i) &= \sum_{t_\ell \in \mathcal{T}^L} d(\mathbf{p}_\ell, \mathbf{W}^\top \mathbf{x}_\ell) \\ &+ \sum_{t_j \in \mathcal{T}^E} d(\mathbf{p}_j, \mathbf{W}^\top \mathbf{x}_j) + \sum_{t_k \in \mathcal{T}^U} d(\mathbf{p}_k, \mathbf{W}^\top \mathbf{x}_k),\end{aligned} \tag{2.4}$$

where by minimizing Eq. (2.4), tPropagation can

- propagate the label information in the propagating process, which is ensured by the first term in Eq. (2.4);
- capture the emoticon information in microblogs since the second term in Eq. (2.4) forces the predicted sentiment of t_j close to $\mathbf{p}_j$ which is fixed to model emoticon information in microblogs; and
- make two processes consistent by forcing the sentiment obtained by these two processes close through the third term in Eq. (2.4).

By adding non-negative constraint on $\mathbf{W}$, we can use $\mathbf{W}_{jk}$ to indicate the importance of the j-th word (d_j) to the k-th sentiment class C_k, which allows us to use the third term to fit the propagation results of words. The third term is to fit the propagation of words, we name it as *wPropagation* and λ_W is employed to control its contribution. We can use a similar way to understand wPropagation in Eq. (2.3). We divide the dictionary $\mathcal{D}$ into two subsets where $\mathcal{D}^L$ denotes words in the lexicons and $\mathcal{D}^U$ is the set of remaining words in $\mathcal{D}$. Then wPropagation in Eq. (2.3) can be rewritten as

$$\sum_{j=1}^{m} d(\mathbf{q}_j, \mathbf{W}_j) = \sum_{d_\ell \in \mathcal{D}^L} d(\mathbf{q}_\ell, \mathbf{W}_\ell) + \sum_{d_k \in \mathcal{D}^U} d(\mathbf{q}_k, \mathbf{W}_k), \tag{2.5}$$

where by minimizing Eq. (2.5), wPropagation can

- incorporate prior knowledge from existing lexicons by forcing the learned sentiment polarity of words close to that in the lexicon by the first term in Eq. (2.5); and
- make two processes consistent by forcing the sentiment polarity obtained by these two processes close.

Compared to traditional text data, microblogging data is informal and noisy. There may exist irrelevant words for sentiment classification, which are difficult to filter during preprocess and may confuse the classifier learning process. For example, some words may be irrelevant to all classes in $\mathcal{C}$, and we should eliminate the their effects, while other

words might be only important to some classes in $\mathcal{C}$. An entity of $\mathbf{W}$, $\mathbf{W}_{jk}$, denotes the importance of the j-th word in the k-th sentiment class C_k. Therefore if a word d_j is totally irrelevant, we should learn a $\mathbf{W}$ whose j-th row is zero; while if d_j is only important to c_k, we should learn a $\mathbf{W}$ whose entities in the j-th row are zero except the k-th entity. These intuitions suggest that $\mathbf{W}$ is likely to be sparse. We add ℓ_1-norm on $\mathbf{W}$ where $\|\mathbf{W}\|_1$ controls the capacity of $\mathbf{W}$ and also ensures that there are many zero entities in $\mathbf{W}$, resulting in a sparse solution. The sparsity of $\mathbf{W}$ is controlled by the parameter λ.

2.3 Our Algorithm With the details of the propagating process and the fitting process, our algorithm is presented in Algorithm 1. We briefly review Algorithm 1 below. In line 1, we use the available labeled data to initialize $\mathbf{W}$, which will be used to initialize $\mathbf{p}_i$ and $\mathbf{q}_j$. From line 2 to line 6, we initialize $\mathbf{p}_i$ according to whether t_i contains emoticons or not, and from line 7 to line 11, we initialize $\mathbf{q}_j$ based on whether d_j is included in the lexicons or not. From line 13 to line 18, we perform the sentiment propagating process. Note that the sentiment of microblogs with emotions and words in the lexicons will be fixed during the propagating process. In line 19, we perform the fitting process and more details to solve $F(\mathbf{W})$ are presented in the following subsection. Finally we use the learned classifier $\mathbf{W}$ to update $\mathbf{p}_i$ and $\mathbf{q}_j$ for the next propagation process. From Algorithm 1, we can see *the significance of alternating the propagating and fitting processes* - the propagating process allows the fitting process to learn from both labeled and unlabeled data with various types of emotional signals, while the fitting process in turn provides supervision information to guide the propagating process.

2.4 Optimizing $F(\mathbf{W})$ In this subsection, we seek a way to optimize $F(\mathbf{W})$. It is easy to verify that $F(\mathbf{W})$ is convex and non-differentiable. $F(\mathbf{W})$ can be rewritten as

$$\begin{aligned} F(\mathbf{W}) &= f(\mathbf{W}) + \lambda\|\mathbf{W}\|_1, \\ f(\mathbf{W}) &= l(\mathbf{W}^\top\mathbf{X}^L, \mathbf{Y}) + \lambda_T \sum_{i=1}^{N} d(\mathbf{p}_i, \mathbf{W}^\top\mathbf{x}_i) \\ &\quad + \lambda_W \sum_{j=1}^{m} d(\mathbf{q}_j, \mathbf{W}_j), \end{aligned} \tag{2.6}$$

where $f(\mathbf{W})$ is convex and differentiable, while $\|\mathbf{W}\|_1$ is convex and non-differentiable, which render the problem non trivial. We adopt the proximal gradient decent method to solve the problem due to its optimal convergence rate [13], which alternates a gradient step and a proximal step,

In the $t+1$-th gradient step,

$$\mathbf{V}_{t+1} = \mathbf{W}_t - \frac{1}{\theta_t} f^{'}(\mathbf{W}_t), \tag{2.7}$$

In the $t+1$-th proximal step,

$$\mathbf{W}_{t+1} = \pi_G(\mathbf{V}_{t+1}), \tag{2.8}$$

Algorithm 1 The Proposed Propagation-based Sentiment Analysis Framework.

Input: $\mathbf{X}$, $\mathbf{Y}$, $\{\alpha, \beta, \lambda_T, \lambda_W, \lambda\}$
Output: The Classifier $\mathbf{W}$

1: Initialize $\mathbf{W} = \arg\min_{\mathbf{W}\geq 0} \; l(\mathbf{W}^\top\mathbf{X}^L, \mathbf{Y})$
2: **if** t_i with emoticon information **then**
3: Initialize $\mathbf{p}_i$ based on its emoticon information
4: **else**
5: Initialize $\mathbf{p}_i = \mathbf{W}^\top\mathbf{x}_i$
6: **end if**
7: **if** d_j in sentiment lexicons **then**
8: Initialize $\mathbf{q}_j$ based on the lexicons
9: **else**
10: Initialize $\mathbf{q}_j = \mathbf{W}_j$.
11: **end if**
12: **while** Not convergent **do**
13: **for** $t_i \in \mathcal{T}^L \cup \mathcal{T}^U$ **do**
14: Update $\mathbf{p}_i = \alpha \frac{\sum_{k=1}^{N} \mathbf{R}_{ik}^{tt}\mathbf{p}_k}{\sum_{k=1}^{N} \mathbf{R}_{ik}^{tt}} + (1-\alpha)\frac{\sum_{k=1}^{m} \mathbf{R}_{ik}^{tw}\mathbf{q}_j}{\sum_{k=1}^{m} \mathbf{R}_{ik}^{tw}}$,
15: **end for**
16: **for** $d_j \in \mathcal{D}^U$ **do**
17: Update $\mathbf{q}_j = \beta \frac{\sum_{k=1}^{m} \mathbf{R}_{jk}^{ww}\mathbf{q}_k}{\sum_{k=1}^{m} \mathbf{R}_{jk}^{ww}} + (1-\beta)\frac{\sum_{k=1}^{N} \mathbf{R}_{jk}^{tw}\mathbf{p}_k}{\sum_{k=1}^{N} \mathbf{R}_{kj}^{tw}}$,
18: **end for**
19: Update $\mathbf{W} = \arg\min_{\mathbf{W}} F(\mathbf{W})$
20: Update $\mathbf{p}_i = \mathbf{W}^\top\mathbf{x}_i$ and $\mathbf{q}_j = \mathbf{W}_j$.
21: **end while**

where $\pi_G(\mathbf{V})$ is the Euclidean projection of $\mathbf{V}$ onto the convex set of G, defined by $\mathbf{W}$ as

$$\pi_G(\mathbf{V}_{t+1}) = \min_{\mathbf{W}\geq 0} \|\mathbf{W} - \mathbf{V}_{t+1}\|_F^2 + \frac{\lambda}{\theta_t}\|\mathbf{W}\|_1. \tag{2.9}$$

It can be further decomposed into m separate sub-problems as

$$\mathbf{W}_{t+1}^i = \min_{\mathbf{W}^i\geq 0} \|\mathbf{W}^i - \mathbf{V}_{t+1}^i\|_2^2 + \frac{\lambda}{\theta_t}\|\mathbf{W}^i\|_1, \tag{2.10}$$

where $\mathbf{W}_{t+1}^i$, $\mathbf{W}^i$ and $\mathbf{V}_{t+1}^i$ are the i-th row of $\mathbf{W}_{t+1}$, $\mathbf{W}$ and $\mathbf{V}_{t+1}$, respectively. It has a closed form solution as below,

$$\mathbf{W}_{t+1}^i = \max(\mathbf{V}_{t+1}^i - \frac{\lambda}{\theta_t}, 0). \tag{2.11}$$

This process can be further accelerated by Nesterov's method [19]. We construct a linear combination of $\mathbf{W}_t$ and $\mathbf{W}_{t+1}$ to update $\mathbf{U}_{t+1}$ as

$$\mathbf{U}_{t+1} = \mathbf{W}_t + \frac{\alpha_t - 1}{\alpha_{t+1}}(\mathbf{W}_{t+1} - \mathbf{W}_t), \tag{2.12}$$

where the sequence $\{\alpha_t\}$ is conventionally set to be $\alpha_{t+1} = \frac{1+\sqrt{1+4\alpha_t^2}}{2}$. More details can be found in [13]. The optimizing algorithm for $F(\mathbf{W})$ is presented in Algorithm 2.

Algorithm 2 Optimizing $F(\mathbf{W})$

Input: $\mathbf{X}^L$, $\mathbf{Y}$, λ, $\alpha_1 = 1$
Output: $\mathbf{W}$

Initialize θ_0 and $\mathbf{W}_1$
while Not convergent **do**
  $\mathbf{U}_t = \mathbf{W}_{t-1} + \frac{\alpha_{t-1}-1}{\alpha_t}(\mathbf{W}_t - \mathbf{W}_{t-1})$
  $\mathbf{V}_{t+1} = \mathbf{U}_t - \frac{1}{\theta_t} f^{'}(\mathbf{U}_t)$
  if $f(\pi_G(\mathbf{V}_{t+1})) > f_{\theta_t}(\pi_G(\mathbf{V}_{t+1}), \mathbf{U}_t)$ **then**
    $\theta_t = \gamma\theta_t$
  end if
  Set $\mathbf{W}_{t+1} = \pi_G(\mathbf{U}_t - \frac{1}{\theta_t} f^{'}(\mathbf{U}_t))$
  Set $\theta_{t+1} = \theta_t$
  Set $\alpha_{t+1} = \frac{1+\sqrt{1+4\alpha_t}}{2}$
  Set $t = t + 1$
end while

In Algorithm 2, $f_{\theta_t}(\mathbf{W}_{t+1}, \mathbf{U}_t)$ is defined as,

$$f_{\theta_t}(\mathbf{W}_{t+1}, \mathbf{U}_t) = f(\mathbf{U}_t) + \langle f^{'}(\mathbf{U}_t), \mathbf{W}_{t+1} - \mathbf{U}_t \rangle + \frac{1}{\theta_t}\|\mathbf{W}_{t+1} - \mathbf{U}_t\|_F^2 \tag{2.13}$$

3 Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework PSA. Follow a common convention [8, 26, 12], we first evaluate the proposed framework with $c = 2$, i.e., $\{negative, positive\}$. Via experiments, we aim to answer the following two questions - (1) how effective is the proposed framework compared with representative sentiment analysis methods; (2) how do different components contribute to the proposed framework PSA. To answer the first question, we compare PSA with various state-of-the-art sentiment analysis methods. To answer the second question, we investigate the effects of different components on PSA. In reality, there may be some microblogs without any sentiment polarity, i.e., neutral, therefore we finally conduct experiments with $c = 3$, i.e., $\{negative, neutral, positive\}$ to assess the capability of the proposed framework in handling multi-class sentiment analysis problem.

3.1 Datasets and Experimental Settings To evaluate the proposed framework, we collect two microblogging datasets. One is written in English with classes { *negative, positive*} and the other is written in Japanese with classes {*negative, neutral, positive* }. Following a common way to preprocess text data for sentiment analysis [24, 23], we employ a unigram model to generate the feature space and the term presence to denote the feature weight. Relations are constructed based on the way introduced in [11]. Below we will present more details about these two datasets.

The first dataset is a microblogging dataset written in English (*eMicroblogs*), which only contains negative and positive microblogs. In [8], the authors collected a set of English microblogs with polarity sentiment labels for supervised sentiment classification. Hu et al. [12] built a correlation graph from the following graph provided by [14]. A widely used English sentiment lexicon, i.e., MPQA opinion Corpus [30], is adopted to provide prior knowledge from the lexicon.

Table 1: Statistics of the Datasets

	eMicroblogs	jMicroblogs
# of Positive Microblogs	11,959	11,701
# of Negative Microblogs	10,303	4,567
# of neutral Microblogs	0	15,863
# of Unigrams	23,346	56,810
# of Pos Lexicon Terms	2,718	2,122
# of Neg Lexicon Terms	4,902	3,195

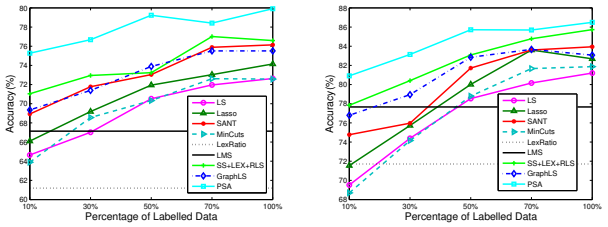
The second dataset is a Japanese microblogging dataset (*jMicroblogs*), which contains negative, neutral and positive microblogs. This dataset is collected using frequent queries, and polarity sentiment labels are manually added to them. Three annotators assign labels to each microblog, and these labels are integrated by taking majority votes to set the final labels. Japanese sentiment lexicon and emoticon lists contain entries collected by crawling web pages as well as in-house data. From this dataset, we extract positive and negative microblogs to form another dataset (*jMicroblogs2*) for the evaluations with $c = 2$.

Some statistics of the datasets are summarized in Table 1. For each dataset, we randomly divide it into two equal parts $\mathcal{A}$ and $\mathcal{B}$, and fix the part $\mathcal{A}$ as the testing set. For $\mathcal{B}$, we choose $x\%$ of $\mathcal{B}$ as the labeled data and the remaining $1 - x\%$ as unlabeled data. For each experiment, we repeat the partition process 5 times and report the average results. In this paper, we vary x as $\{10, 30, 50, 70, 100\}$. Note that when $x = 10$, we actually choose 5% of the whole dataset as labeled data and 45% of the whole dataset as unlabeled data. We choose a common metric accuracy to evaluate the performance of sentiment analysis [24, 11].

In $F(\mathbf{W})$, $l(\cdot,\cdot)$ is a loss function and $d(\cdot,\cdot)$ is a distance metric. We choose square loss for $l(\cdot,\cdot)$ and Euclidean distance for $d(\cdot,\cdot)$ in this work. Note that our evaluations will first focus on $c = 2$ with *eMicroblogs* and *jMicroblogs2* datasets, and finally we assess the capability of the proposed framework in handling multi-class sentiment analysis problem with *jMicroblogs*.

3.2 Performance Evaluation To answer the first question, we choose representative methods from three groups as baseline methods. The first group includes representative supervised methods:

- *LS:* This method performs least square, a widely used supervised method, on the labeled microblogs.
- *Lasso:* Lasso can force some learned coefficients to exact zero, and then excludes the corresponding terms (irrelevant terms) from the sentiment classifier [29]. Lasso works on the labeled microblogs only.
- *SANT:* SANT is based on Lasso, but exploits the correlation information among microblogs to facilitate senti-



(a) *eMicroblogs* (b) *jMicroblogs2*

Figure 2: The Performance Comparison

ment analysis [12]. It works on the labeled microblogs with their correlation relations.

- *MinCuts:* This method utilizes contextual information via the minimum cut framework for sentiment classification [22].

Note that we choose square loss as the loss function, therefore the baseline methods *LS*, *Lasso*, and *SANT* are based on square loss. We do not compare our method with methods based on hinge loss such as SVM, and logistic loss such as logistic classifier, since we can extend the proposed framework with hinge loss and logistic loss.

Methods in the second group are unsupervised methods based on lexicon

- *lexRatio:* This method aggregates the sentiment polarity of terms in a microblog based on a lexicon to determine its sentiment orientation [30].

- *LMS:* LMS is an augmented lexicon-based method. LMS first utilizes the lexicon and other rules to obtain the labels of microblogs, and then trains a classifier based on the microblogs labeled by the lexicon-based method [31].

Methods in the third group are representative semi-supervised methods,

- *SS+LEX+RLS:* SS+LEX+RLS incorporates lexical information and unlabeled data within standard regularized least squares [25].

- *GraphLS:* Graph regularization is one of the most popular techniques to perform semi-supervised learning [34], and it assumes that unlabeled and labeled data follow the same distribution, and that similar microblogs should have similar sentiment polarity.

The parameters of all baseline methods are determined via cross-validation. For PSA, we set $\{\lambda_T = 0.1, \lambda_W = 0.3, \lambda = 0.001\}$ and $\{\lambda_T = 0.5, \lambda_W = 0.5, \lambda = 0.01\}$ in *eMicroblogs* and *jMicroblogs2*, respectively. We will provide more details about parameter analysis for PSA in the following subsections. The comparison results are shown in Figure 2.

We make the following observations:

- With the increase of labeled data, the performance of supervised methods and semi-supervised methods increases, while the performance of unsupervised methods is independent of the number of available labeled data. We also note that when the number of labeled data is small, semi-supervised methods outperform supervised methods, and when the number of labeled data is large enough, supervised methods can perform better than unsupervised lexicon-based methods.

- *Lasso* performs better than *LS*, which indicates the existence of irrelevance words for sentiment analysis and suggests that excluding irrelevant terms from microblogs can improve the sentiment classification performance. *SANT* obtains better performance than Lasso. The only difference between *SANT* and *Lasso* is that *SANT* exploits microblog-microblog correlation information, which demonstrates that microblog-microblog correlation information contains complementary information to the content of microblogs.

- *SS+LEX+RLS* outperforms *GraphLS*. One major reason is that *SS+LEX+RLS* incorporates lexicons, which demonstrates the importance of prior knowledge from lexicons.

- The proposed framework PSA always obtains the best performance. Compared to the best performance of baseline methods, PSA gains 5.95% with 10% and 8.38% with 70% relative improvement in *eMicroblogs*. We apply t-test to compare the performance of PSA and the best performance of baseline methods. The t-test results demonstrate that our semi-supervised framework with sentiment propagation achieves significant performance improvement (with the significance level 0.01). There are two major reasons - (1) *SS+LEX+RLS* and *GraphLS* do not include components to deal with irrelevant words, while PSA can reduce the influence of irrelevant words on the classifier via sparse coding; (2) short microblog content cannot provide sufficient information to indicate the similarities between microblogs for *SS+LEX+RLS* and *GraphLS*, while PSA provides the propagation process to incorporate various types of emotional signals such as correlations, emoticons, and lexicons. We will present more details about the contributions of components of PSA in the following subsection.

With evidence above, we can draw an answer to the first question - the proposed framework PSA gains significant performance improvement for sentiment classification compared to various representative baseline methods.

3.3 Impact of Different Components on PSA In the last section, we demonstrate the effectiveness of PSA. In this subsection, we investigate the effects of different components on PSA correspondingly to answer the second question. There are three key components of PSA: tPropagation, wPropagation and the sparse coding component to handle irrelevant words, controlled by parameters $\{\lambda_T, \lambda_W, \lambda\}$, respectively. By setting one, two, or three of these parameters

Table 2: The Performance of PSA by Systematically Eliminating its Components; Note that "Optimal" in the table denotes a non-zero optimal value of the corresponding parameter, and "loss" represents the performance reduction compared to PSA with all three components.

λ_T	λ_W	λ	eMicroblogs(loss)		jMicroblogs2(loss)	
			10%	100%	10%	100%
Optimal	Optimal	Optimal	$75.27(N.A.)$	$79.91(N.A.)$	$80.91(N.A.)$	$86.41(N.A.)$
0	Optimal	Optimal	$69.95(-7.07\%)$	$76.01(-4.88\%)$	$75.07(-7.22\%)$	$84.31(-2.43\%)$
Optimal	0	Optimal	$71.73(-4.70\%)$	$75.32(-5.74\%)$	$76.51(-5.44\%)$	$83.89(-2.92\%)$
Optimal	Optimal	0	$73.84(-1.90\%)$	$77.39(-3.15\%)$	$78.44(-3.05\%)$	$85.83(-0.67\%)$
0	Optimal	0	$67.94(-9.74\%)$	$75.08(-6.04\%)$	$72.23(-10.73\%)$	$83.77(-3.06\%)$
0	0	Optimal	$66.10(-12.18\%)$	$74.15(-7.21\%)$	$71.55(-11.57\%)$	$82.69(-4.31\%)$
Optimal	0	0	$69.03(-8.29\%)$	$74.79(-6.41\%)$	$74.91(-7.42\%)$	$83.02(-3.92\%)$
0	0	0	$64.65(-14.11\%)$	$72.64(-9.10\%)$	$69.52(-14.08\%)$	$81.20(-6.03\%)$

to zero, we can systematically eliminate the effect(s) of one, two or three components from the proposed framework. The results of PSA with different components are shown in Table 2 and we only show performance with 10% and 100% since we have similar observations with other settings. Note that "loss" represents the performance decrease compared to PSA with all three components; "Optimal" in the table denotes a non-zero optimal value of the corresponding parameter via cross-validation, and "0" denotes the corresponding parameter with the value zero. For example, the second row represents the performance of PSA without tPropagation, while the last row represents the performance without any components, which is actually the baseline method *LS*.

The second to the fourth rows in Table 2 denote the performance when we eliminate the effect of one component from PSA. We note that performance consistently degrades after eliminating any of these components, which suggests that these three components contain complementary information to each other and are useful for the proposed framework. We also note that removing different components may result in very different performance reduction, which indicates that the contributions of different components to PSA may differ. For example, when the number of labeled data is small, it seems that tPropagation plays a more important role in PSA; hence, eliminating tPropagation decreases the performance most for both datasets.

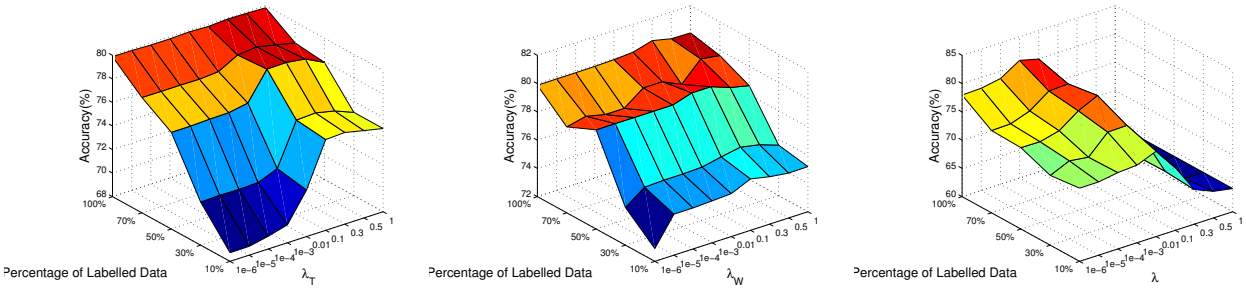
The performance when we eliminate the effects of two components from PSA is shown from the fifth to the seventh rows in Table 2. The performance further decreases compared to that when eliminating only one component. When eliminating the effects of all three components, the performance is worst as presented in the eighth row in Table 2. Compared to PSA with all components, the performance of PSA without any components decreases 14.08% with 10% and 9.10% with 100% in *jMicroblogs2*.

3.4 Parameter Analysis In this subsection, we conduct parameter analysis for PSA. There are three important parameters for PSA - (1) λ_T controlling the contribution from tPropagation, (2) λ_W controlling the contribution from wPropagation, and (3) λ controlling the capability of handling noise data. Hence we study the effect of each of the three parameters by fixing the other 2 to see how the performance of PSA varies with different percentages of labeled data. The values of $\{\lambda_T, \lambda_W, \lambda\}$ are varied as $\{$1e-6,1e-5,1e-4,1e-3,0.01,0.1,0.3,0.5,1$\}$. The processes of parameter analysis for *eMicroblogs* and *jMicroblogs2* are similar, and we present the details for *eMicroblogs* to save space. Examples of experimental results are presented next.

To study the effect of λ_T on PSA, we fix $\{\lambda_W = 0.3, \lambda = 0.001\}$. The performance variation w.r.t. λ_T and percentages of labeled data is depicted in Figure 3(a). When the number of labeled data is small, the performance of PSA is very sensitive to λ_T. For example, when we choose 10% labeled data, the performance increases a lot with λ_T from 0.01 to 0.1, which suggests the importance of tPropagation to PSA when a small number of labeled data is available. Generally with the increase of λ_T, the performance trends to dramatically increase and then gradually decrease, and in a certain region, the performance seems stable such as λ_T in $\{0.1, 0.3, 0.5\}$. These patterns can be utilized to determine the optimal value of λ_T in practice.

By setting $\{\lambda_T = 0.1, \lambda = 0.001\}$, the performance variation w.r.t. λ_W and percentages of labeled data is demonstrated in Figure 3(b). Compared to λ_T, the performance is less sensitive to λ_W. Generally, with the increase of λ_W, the performance first increases, and after a certain value, the continued increase of the value of λ leads to performance reduction. In an extreme case, when λ_W is $+\infty$, wPropagation will dominate the learning process and the performance is mainly controlled by the lexicon.

The performance variation in terms of λ and percentages of labeled data is shown in Figure 3(c) with $\{\lambda_T = 0.1, \lambda_W = 0.3\}$. λ controls the sparsity of $\mathbf{W}$. From $\lambda = 1e$-6 to $\lambda = 1e$-3, the performance increases a lot. These results support that eliminating irrelevant features can significantly improve the performance. When λ continues to increase, the sparse coding part dominates the learning process and $\mathbf{W}$ becomes sparser. This may exclude many useful words and results in performance reduction. For example, when λ is $+\infty$, the learned $\mathbf{W}$ is zero, which eliminates all words.



(a) λ_T (b) λ_W (c) λ

Figure 3: Parameter Analysis for The Proposed Framework PSA.

Table 3: Performance Comparison for Multi-Class Sentiment Analysis.

Algorithms	10%	30%	50%	70%	100%
LS	55.88	56.99	58.08	58.97	59.40
Lasso	56.42	58.06	58.93	59.75	60.20
SANT	56.78	58.42	59.37	59.81	60.14
MinCuts	55.92	57.39	58.19	59.03	59.38
SS+LEX+RLS	59.51	60.94	61.73	62.56	62.92
GraphLS	59.09	59.97	60.18	61.85	62.02
PSA	63.83	64.34	64.97	65.01	65.67

3.5 PSA for Multi-Class Sentiment Analysis In reality there might be many microblogs without any sentiment polarity such as neutral microblogs. It is important to assess the capability of the proposed framework PSA in handling the problem of multi-class sentiment analysis. In this subsection, we compare PSA with representative baseline methods in *jMicroblogs* which contains negative, neutral and positive microblogs, and the results are shown in Table 3. The accuracies of *lexRatio* and *LMS* are 51.79% and 53.27%, respectively. Since their performance is independent on the training set, we do not show their results in the table.

Observations with multi-class data are similar to those of binary-class data. With the increase of training sets, the performance of supervised and semi-supervised methods increases. Most of the time, semi-supervised methods outperform supervised methods especially when the training set is small. The proposed framework always obtains the best performance and the t-test results show that the improvement is significant. These results suggest that the proposed framework PSA can address the multi-class sentiment analysis problem effectively.

4 Related Work

Sentiment analysis methods fall into three categories, i.e., supervised methods, unsupervised methods and semi-supervised methods [23, 15]. Supervised methods first extract a feature space to represent the data and then train a sentiment classifier from manually labeled training data (or a mapping function from the feature space to the labels) [24, 22]. Lexicon-based methods are the most representative methods for unsupervised sentiment analysis, which determine the sentiment polarity of a given document by predefined sentiment lexicons [27]. There are three common ways to build sentiment lexicons. The first is to manually label the sentiment polarity of a set of representative words such as MPQA [30]. The second is to learn sentiment orientation of a word from its semantically/linguistically related words with the help of dictionaries (e.g., WordNet) [2]. The third is to infer sentiment orientation of words from a given corpus by exploring the relation between the words and some observed seed sentiment words [18, 32]. Semi-supervised methods take advantage of both labeled and unlabeled data and can rapidly adapt to new domains with less human efforts [25].

Recently, microblogging services became important resources to understand public opinions, and sentiment analysis for microblogging data has attracted increasing attention. In [8], a framework is proposed to obtain a sentiment classifier with distant supervision. Authors in [4] explore the meta information of words and the linguistic characteristics of tweets for sentiment analysis. Zhang et al. propose to combine lexicon-based and learning-based methods for Twitter sentiment analysis [31]. Except content information, microblogging data provides extra information related to its sentiment polarity such as social relations and emoticons. This extra information is strongly correlated with sentiment polarity. In [12], the authors find strong evidence to support that (1) the sentiment polarity of tweets from the same users or two connected users are more likely to be similar than that of randomly chosen tweets; (2) the sentiment polarity of tweets is likely to be consistent with that of emoticons in tweets. There are some efforts to exploit extra information presented in microblogging data to help sentiment analysis. For example, [26, 28, 12] investigate how to incorporate social relations for sentiment analysis, while [33] exploit emoticons to improve sentiment analysis for microblogging data. Our proposed framework is substantially different from these methods. First our framework is semi-supervised; methods while above methods are either supervised methods or unsupervised methods. Second, our framework provides a unified way to incorporate extra sources such as social relations and emoticons into a coherent model via the propagating process, while above methods are developed for a

certain type of extra information, i.e., either social relations or emoticons.

5 Conclusion

In this paper we propose a novel propagation-based sentiment analysis framework PSA which alternates a propagating process and a fitting process. The fitting process learns a classifier to fit label information and propagation results from the propagating process, and delivers the classification results as supervision information to guide the propagating process, while the propagating process in turn propagates classification results from the fitting process and returns the propagation results to the fitting process, which allows the fitting process to learn from unlabeled data. The experimental results on real-world microblogging datasets demonstrate the effectiveness of the proposed framework. Since the proposed propagation-based framework is a general learning framework, we will apply this framework for other applications such as spam detection in the future.

Acknowledgments

This material is based upon work supported by, or in part by, the U.S. Army Research Office (ARO) under contract/grant number 025071, and the Office of Naval Research(ONR) under grant number N000141010091.

References

[1] R. P. Abelson. Whatever became of consistency theory? *Personality and Social Psychology Bulletin*, 1983.

[2] A. Andreevskaia and S. Bergler. Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*, 2006.

[3] S. Asur and B. A. Huberman. Predicting the future with social media. In *WI-IAT*, 2010.

[4] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *ACL*, 2010.

[5] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011.

[6] Y. Chang, L. Tang, I. Yoshiyuki and L. Yang. What is tumblr: A statistical overview and comparison. *ACM SIGKDD Explorations Newsletter*, 2014.

[7] S. Chang, G. Qi, C. Aggarwal, J. Zhou, M. Wang and T. Huang. Factorized Similarity Learning in Networks. *ICDM*, 2014.

[8] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009.

[9] E. Hatfield, J. T. Cacioppo, and R. L. Rapson. Emotional contagion. *Current Directions in Psychological Science*, 1993.

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

[11] X. Hu, J. Tang, H. Gao, and H. Liu. Unsupervised sentiment analysis with emotional signals. In *WWW*, 2013.

[12] X. Hu, L. Tang, J. Tang, and H. Liu. Exploiting social relations for sentiment analysis in microblogging. In *WSDM*, 2013.

[13] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, 2009.

[14] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, 2010.

[15] B. Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2010.

[16] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman and Hall/CRC, 2007.

[17] K.-L. Liu, W.-J. Li, and M. Guo. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, 2012.

[18] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *WWW*, 2011.

[19] Y. Nesterov and I. E. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

[20] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 2010.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

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

[23] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

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

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

[26] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, 2011.

[27] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[28] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *KDD*, 2011.

[29] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1996.

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

[31] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu. Combining lexicon based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89, 2011.

[32] L. Zhang and B. Liu. Identifying noun product features that imply opinions. In *ACL (Short Papers)*, 2011.

[33] J. Zhao, L. Dong, J. Wu, and K. Xu. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *KDD*, 2012.

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *NIPS*, 2004.