9-2014

Build emotion lexicon from microblogs by combining effects of seed words and emoticons in a heterogeneous graph

Kaisong SONG

Shi FENG

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Daling WANG

Ling CHEN

See next page for additional authors

Author

Kaisong SONG, Shi FENG, Wei GAO, Daling WANG, Ling CHEN, and Chengqi ZHANG

Build Emotion Lexicon from Microblogs by Combining Effects of Seed Words and Emoticons in a Heterogeneous Graph

Kaisong Song[1], Shi Feng[1,2], Wei Gao[3], Daling Wang[1,2], Ling Chen[4], Chengqi Zhang[4]

[1]School of Information Science and Engineering, Northeastern University, Shenyang, China
[2]Key Laboratory of Medical Image Computing (Northeastern University), Ministry of Education, China
[3]Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar
[4]Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia
kaisongsong@gmail.com, {fengshi, wangdaling}@ise.neu.edu.cn
wgao@qf.org.qa, {ling.chen, chengqi.zhang}@uts.edu.au

ABSTRACT

As an indispensable resource for emotion analysis, emotion lexicons have attracted increasing attention in recent years. Most existing methods focus on capturing the single emotional effect of words rather than the emotion distributions which are helpful to model multiple complex emotions in a subjective text. Meanwhile, automatic lexicon building methods are overly dependent on seed words but neglect the effect of *emoticons* which are natural graphical labels of fine-grained emotion. In this paper, we propose a novel emotion lexicon building framework that leverages both seed words and emoticons simultaneously to capture emotion distributions of candidate words more accurately. Our method overcomes the weakness of existing methods by combining the effects of both seed words and emoticons in a unified three-layer heterogeneous graph, in which a multi-label random walk (MLRW) algorithm is performed to strengthen the emotion distribution estimation. Experimental results on real-world data reveal that our constructed emotion lexicon achieves promising results for emotion classification compared to the state-of-the-art lexicons.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Dictionaries, Thesauruses*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Web 2.0*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

Keywords

emotion lexicon; heterogeneous graph; microblogs; emoticon; seed word

1. INTRODUCTION

Nowadays, more and more people are willing to express their attitudes and feelings in social media such as Twitter[1] and Weibo[2] rather than just passively browse and receive information. With the prolific rise of user-generated content in social media, how to effectively analyze users' sentiments has received much attention in the past decade. Emotion lexicons, which annotate words with their expressed emotions, are crucial to the success of sentiment analysis. Therefore, building high-quality emotion lexicons becomes essential for many different kinds of sentiment analysis applications [12, 13, 22, 10, 11, 7, 24].

A lot of works have been done for building emotion lexicons [4, 6, 8, 18], where each word is given a positive or negative label automatically. However, the binary representation of emotion may be oversimplified. For example, rather than simply assigning a negative emotion label to the word "self-abasement", it is more accurate to annotate the expressed emotion as sadness and disgust. Therefore, studies on constructing emotion lexicons that assign the entry words into fine-grained categories of emotions, such as *happiness*, *like*, *disgust*, *sadness*, and *anger*, have emerged recently [29, 30, 34]. In addition, for many sentiment analysis applications, it is beneficial to know not only the binary or multiple emotion classes of a word, but also the emotion intensity. For example, how favorably or unfavorably do people feel about a new product, movie or a TV show. As a result, several recent lexicons [3, 5, 25, 30] associate words with both emotion classes and corresponding valence scores to represent emotion intensity. In this paper, we propose a more generalized solution that derives the emotion distributions of entry words. That is, for each entry word, we estimate its probabilities belonging to various emotion classes.

Traditional lexicon construction overly depends on the seed emotion words selected from large set of words [28, 32], but ignores the emotion of the entire post which accommodates the entry words. We observe that many microblog posts are accompanied by abundant *emoticons* that naturally convey the overall sentiment of the post. It is thus expected that emoticons play complementary roles with seed words in building fine-grained emotion lexicon. For instance,

---

[1]http://www.twitter.com/
[2]http://www.weibo.com/

let us consider the following two example microblogs[3]:

(1) I must **accuse** myself. So *sad*, her injury was my fault...

(2) He must be **accused**. Her injury was his fault! 😬

where *sad* is a seed word and **accuse** is the candidate emotion word. To annotate the emotion of **accuse**, if we consider its co-occurrence with seed word only, we may infer from the first post that **accuse** is associated with the emotion class *sadness*. However, considering its occurrence in the second post, which conveys an overall emotion of *anger* represented by the emoticon 😡, we may also assign **accuse** to the emotion class *anger*. Hence, considering either seed words or emoticons may produce incomplete and inaccurate representation of sentiment. Our idea is to combine the effects of both seed words and emoticons using a unified method, which captures the emotion of candidate words from different perspectives.

Intuitively, the more frequently a candidate emotion word co-occurs with seed words and emoticons bearing specific emotions, the more likely the candidate word converges to the real emotion distribution. In this paper, we propose a novel and extensible emotion lexicon building method by formulating the lexicon building as a multi-label random walk (MLRW) problem for estimating the emotion distributions of candidate words. The candidate words with probabilities of some main emotion classes greater than a specified threshold are selected into lexicon. Our contributions are three-fold:

- We propose a unified framework that combines the effects of seed words and emoticons co-occurring with candidate words in microblogs, which captures the fine-grained emotion of candidate words more accurately.

- We develop an effective multi-label random walk algorithm for emotion distribution estimation based on a three-layer heterogeneous graph, where vertices at different layers represent emoticons, seed words and candidate words jointly and the edges represent their corresponding co-occurrence relationships.

- We conduct sentence-level emotion classification experiments on a real-world microblog dataset using our constructed emotion lexicon, which shows promising performance in single-label and multi-label classification compared to emotion lexicons built by state-of-the-art approaches. We make our learned lexicon public accessible.

The remainder of this paper is organized as follows: In Section 2, we introduce related work on building emotion lexicons and their applications to sentiment analysis; we describe general inference method for finding the emotion distributions of emoticons in Section 3 and seed words selection method in Section 4; in Section 5, we present our effective multi-label random walk algorithm for emotion distribution estimation based on a three-layer heterogeneous graph; Section 6 provides the detailed results of evaluation; we conclude our work and give future directions in Section 7.

2. RELATED WORK

It is important to build high-quality emotion lexicons for sentiment analysis tasks. Most of the existing methods focus

on building coarse-grained emotion lexicons where each entry is assigned with a positive or negative sentiment label [4, 6, 8, 14, 18]. In contrast to such lexicons, methods of building fine-grained emotion lexicon have also been proposed [20, 23, 30]. Our work fits in the latter category for building a fine-grained emotion lexicon. In addition, an emotion word conveying multiple emotions may associate different intensity with each emotion. Similar to Staiano et al. [25], we represent emotion words in our lexicon as emotion distributions such that for a given emotion word, the score in each emotion dimension represents the emotion strength.

Existing methods build emotion lexicons largely from a set of selected seed emotion words. Strapparava and Valitutti [26] created WordNet-Affect, an affective extension of WordNet, by leveraging seed words to make all WordNet synonyms have the same emotion. Xu et al. [30] used pointwise mutual information (PMI) to measure the correlation strength between seed words and candidate words. Xu et al. [28] adopted a graph-based algorithm which allows candidate words to learn emotion from connected seed words. Yang et al. [32] proposed an emotion-aware LDA model to build a domain-specific emotion lexicon using a minimal set of domain-independent seed words. All these approaches stem from the basic intuition that the emotion of a candidate word is determined by its frequently co-occurring seed words. However, they heavily rely on the subjective selection of seeds, rendering the low coverage of the constructed lexicons. Moreover, they ignore the overall emotion of the entire post which cannot be fully reflected by the selected seed words.

Graphical emoticons naturally reflect the emotion of a user and the post as a whole. Some studies [23, 34] even leveraged people's affective tags for online news articles as annotations. Zhao et al. [35] proposed a microblog emotion classifier trained on posts using emoticons as the ground truth. Yang et al. [31] adopted a variation of PMI to measure the similarity between emoticons and candidate words. Further, Feng et al. [6] integrated emoticons and candidate words into a graph model to mutually reinforce the ranking of candidate words. Unlike seed words, emoticon may convey multiple complex emotions, which is inherently versatile. Nevertheless, emoticons are generally noisy and the embedded complex emotions sometimes are hard to differentiate. In this paper, we model the sentiment as an emotion distribution and aim to strengthen the estimation of emotion distribution of candidate words by combining the effects of seed words and emoticons in a unified framework. A multi-label random walk algorithm is proposed to capture the emotion distribution accurately.

Emotion lexicons have been used by all kinds of sentiment analysis and other related applications [12, 13, 22, 10, 11, 7, 24]. While we focus on building a high-quality fine-grained emotion lexicon, we resort to a basic voting-based sentiment classification method to assess the quality of lexicon generated by our proposed method.

3. EMOTION DISTRIBUTION OF EMOTICONS

Emoticons are commonly contained in microblog posts which are of rich quantity and can reflect users' overall emotion in the post explicitly [1, 31, 35]. Unlike emoticons, seed emotion words are selected in a subjective manner

Table 1: Polarity vs. fine-grained emotion

Sentiment Polarity	Emotion Category
Positive	happiness, like, surprise
Negative	disgust, sadness, anger, fear

and usually convey the most intensive emotion rather than overall emotion. Further, as opposed to the binary emotion polarity, an emoticon can express complex emotions which in nature can be modeled as a distribution of different e-motions. To capture such fine-grained emotion types, we first introduce an inference method for finding the emotion distribution for emoticons in this section, and then the inferred distribution is leveraged to guide building the emotion lexicon by incorporating seed emotion words for further distinguishing similar emotions.

3.1 Building Emoticon Dataset

We make use of a publicly available microblog dataset named NLPCC2014 corpus[4] (see Section 6.1 for details) to train an emotion classifier for inferring the emotion distribution of emoticons. In this corpus, human annotators identify the emotion-bearing sentences in the microblogs and annotate each identified sentence with one or two emotion labels that reflect the major (for the one-label case) or the major and secondary emotions (for the two-label case) of the sentence. To build our training data, we extract a subset with 3,232 sentences that all contain emoticons by discarding the pure text sentences in the corpus. Each sentence *sen* in this subset is preprocessed into the following form:

$$sen = [\{emc_1, emc_2, ..., emc_N\}^*, emo]$$

where $\{emc_1, emc_2, ..., emc_N\}$ is the full set of N emoticons in our system, $\{...\}^*$ is a subset of it meaning the specific emoticons contained in the sentence and *emo* is the emotion label. Note that if a sentence has two emotion labels, we split it into two instances each containing a unique emotion label. As a result, we build a training set with 3,600 sentences.

3.2 Inferring Emotion Distribution

Here we aim to estimate the emotion distribution for each emoticon. Let C denote the possible label for *emo* and A denote the possible emoticon in the full set of emoticons. Given an emoticon, we can obtain the probability of its classified emotion based on Baysian rule:

$$P(C = c_j | A = a_i) = \frac{P(C = c_j) P(A = a_i | C = c_j)}{\sum_j P(C = c_j) P(A = a_i | C = c_j)}$$

where a_i denotes the i-th emoticon emc_i of A (i.e., $i = 1...N$) and c_j is the j-th emotion of C. Based this formula, we represent each emoticon as a f-dimensional vector of emotion distribution $v = <P_1, ..., P_f>$ (so $j = 1...f$). Specifically, we adopt the popular seven-level categorization of fine-grained emotion following [27, 30], which are described in Table 1 (therefore $f = 7$).

Based on the results of inference, we demonstrate 32 most frequently used emoticons and their emotion distributions in Figure 1. We can observe that some emoticons convey only a single prominent emotion like 🙇, but many of them
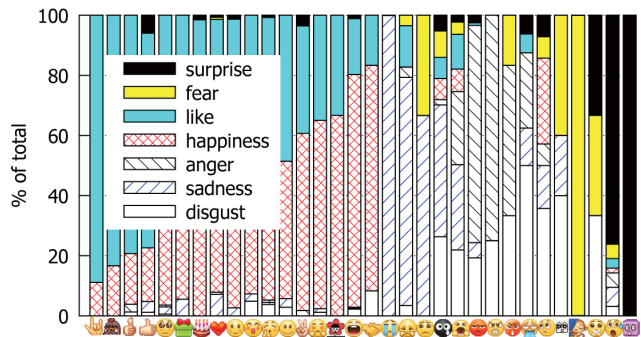

Figure 1: Emotion distribution of emoticons

suggest multiple complex emotions like 😫, and some others such as 😎 have a major emotion *anger* and a secondary emotion *disgust*. We expect that the emotions of such different granularity from emoticons can be used to help capture the emotion distribution of candidate words.

4. SEED EMOTION WORDS

Although emoticons are natural and versatile, there is at times subtle difference among emoticons, such as those containing emotions of anger and sadness (see Figure 1). In addition, emoticons are noisy sentiment labels [9]. Therefore, only using emoticons for selecting candidate words would be suboptimal in the sense that their effect on distinguishing similar emotions of candidate words in fine-grained level is problematic. As mentioned earlier, the seed emotion words typically bear some salient emotion, which can provide benefit to distinguish the subtle difference among the emotions of candidate words resulting from the co-occurring emoticons.

In this paper, we adopt a semi-automatic approach to choose seed words and ensure that the selected seed words can have straightforward impact on the co-occurring candidate words. Specifically, we rank all the words in the microblog dataset (see Section 6.1) according to their occurrence frequency; For those high-frequency words, we cross-reference the entries of a state-of-the-art manually created emotion lexicon from DUTIR group[5] called EWN (i.e., Emotion Words Noumenon) [27, 30], which provides seven possible emotions (i.e., happiness, like, surprise, disgust, sadness, anger, fear), five level of emotion intensity (i.e., 1, 3, 5, 7, 9) and no more than two emotion labels for each entry word. For each emotion, we manually select five high-frequency words with strong intensity as seed words. To represent the selected seed words, we adopt the same vector representation as that of emoticons. The only difference is that the probability value of the element is replaced by the ratio of the corresponding emotion's intensity with respect to all possible emotions' intensity.

5. BUILDING EMOTION LEXICON

We aim to overcome the weakness of relying on either seed words or emoticons alone: although seed words have strong indicative and discriminative capacity, seeds selection is subjective, which cannot induce to fully cover fine-grained emotions of candidate words; emoticons are finer grained, but

---

[4] http://tcci.ccf.org.cn/conference/2014/index.html

[5] http://ir.dlut.edu.cn/
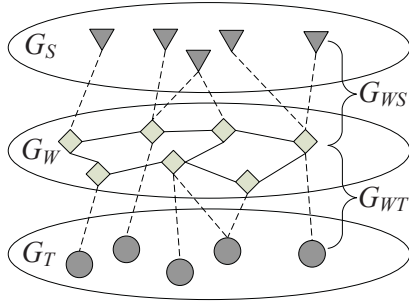
Figure 2: Our heterogeneous graph, where circles denote emoticons, diamonds denote candidate words, triangles denote seed words, solid lines and dotted lines correspond to different types of edges

are prone to noise and less discriminative. In this section, we try to combine their effects to infer the emotion distribution of candidate words more accurately. Specifically, we build a unified framework based on a three-layer heterogeneous graph, where nodes at each layer corresponds to seed words, emoticons and candidate words. Then we propose a multi-label random walk (MLRW) algorithm to strengthen the emotion distribution of candidate words resulting from complementary effect of seed words and emoticons. Finally, we output candidate words with the resulted emotion distribution as a lexicon.

5.1 Symbols and Notations

We first introduce some notations which are frequently used later. Let $W = \{w_1, w_2, ..., w_{|W|}\}$ be a candidate word set, $S = \{s_1, s_2, ..., s_{|S|}\}$ be a seed word set, and $T = \{t_1, t_2, ..., t_{|T|}\}$ be an emoticon set. For any element in W, T and S, we represent it as a vertex v in the graph. So we have three vertex sets V_W, V_T and V_S, and the entire vertex set $V = V_W \cup V_T \cup V_S$. Each $v \in V$ is a f-dimensional vector of emotion distribution $v = <P_1, ..., P_f>$ and $\sum_{i=1}^f P_i = 1$. If a vertex v_i often co-occurs with v_j, then there will be an edge e_{ij} between them. Let $E_{WW} = \{e_{ij}|i, j \in W, i \neq j\}$ denote the edges between candidate words, $E_{WT} = \{e_{ij}|i \in W, j \in T\}$ as edges between candidate words and emoticons, and $E_{WS} = \{e_{ij}|i \in W, j \in S\}$ as edges between candidate words and seed words. So edge set $E = E_{WW} \cup E_{WT} \cup E_{WS}$. As a result, the dataset is formulated as a heterogeneous graph $G = (V, E)$, as illustrated in Figure 2. Emoticons and seed words are priori knowledge, so V_S and V_T are fixed. Our goal is to estimate the vector of emotion distribution for V_W.

5.2 Building Heterogeneous Graph

We assign the edge weights for E_{WW}, E_{WS} and E_{WT} on the graph. Let **WW**, **WS** and **WT** be the adjacent matrices of subgraphs $G_W = (V_W, E_{WW})$, $G_{WS} = (V_W \cup V_S, E_{WS})$ and $G_{WT} = (V_W \cup V_T, E_{WT})$, respectively. We resort to a variant of pointwise mutual information (PMI) [19] for measuring the correlation between nodes. The weight of edge e_{ij} in G_W is defined as follow:

$$\mathbf{WW}_{ij} = \begin{cases} c(w_i, w_j) * \delta\left(\frac{P(w_i, w_j)}{P(w_i)P(w_j)}\right) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where $\delta(x) = \frac{1}{1+e^{-x}}$ is a logistic function instead of logarithm function in PMI (logistic function ensures all the elements in the vector of emotion distribution are positive), $P(w) = \frac{|M_w|}{|M|}$ is the probability of candidate word w occurring in the entire microblog set M (M_w is the microblogs contain w), $c(w_i, w_j) = |M_{w_i} \cap M_{w_j}|$ is co-occurrence count of w_i and w_j (M_{w_i} and M_{w_j} are the microblogs containing words w_i and w_j, respectively), and $P(w_i, w_j) = \frac{c(w_i, w_j)}{|M|}$ is the probability that w_i and w_j co-occur. Similarly, we can also define adjacent matrix **WS** and **WT** as follow:

$$\mathbf{WS}_{ij} = c(w_i, s_j) * \delta\left(\frac{P(w_i, s_j)}{P(w_i)P(s_j)}\right)$$

$$\mathbf{WT}_{ij} = c(w_i, t_j) * \delta\left(\frac{P(w_i, t_j)}{P(w_i)P(t_j)}\right)$$

where $P(s) = \frac{|M_s|}{|M|}$ is the probability of seed word s occurring in M, $P(t) = \frac{|M_t|}{|M|}$ is the probability that emoticon t occurs, $P(w_i, s_j) = \frac{c(w_i, s_j)}{|M|}$ is the probability that w_i and s_j co-occur, and $P(w_i, t_j) = \frac{c(w_i, t_j)}{|M|}$ is the probability that w_i and t_j co-occur. Note that our variant of PMI is aimed for generating positive edge weights. This is necessary to ensure the probabilities in the generated emotion distribution to be non-negative (see Section 5.3).

5.3 Multi-label Random Walk (MLRW)

We propose the MLRW algorithm over the above undirected heterogeneous graph. Previous variants of PageRank-based methods [6, 36] for building emotion lexicon are aimed for ranking graph nodes all with single values, and the ranking scores of nodes could become negative during iterations. This is not directly applicable to our case. We intend to generate (and update) the emotion probability distribution for each node based on the distributions of other nodes and the edge weights, rather than just ranking candidate words. Meanwhile, we need normalization in adjacent matrices to ensure the probabilistic distribution for candidate words in each iteration. We adopt the form of power iteration formula [17] for updating the distribution in each round:

$$\mathbf{P}_W^{(k+1)} = (1-\alpha)\mathbf{e} + \alpha\mathbf{X}_W^{(k)}, \quad and$$

$$\mathbf{X}_W^{(k)} = \beta_1 \widetilde{\mathbf{WW}}\mathbf{P}_W^{(k)} + \beta_2 \widetilde{\mathbf{WT}}\mathbf{P}_T^{(k)} + (1-\beta_1-\beta_2)\widetilde{\mathbf{WS}}\mathbf{P}_S^{(k)}$$

where $\mathbf{P}_W : |W| \times f$, $\mathbf{P}_T : |T| \times f$ and $\mathbf{P}_S : |S| \times f$ are output matrices of candidate words, emoticons and seed words, respectively, $\mathbf{e}$ is an identity vector and α is damping factor [2], β_1 and β_2 are adjustable relative weights, $\widetilde{\mathbf{WW}}$, $\widetilde{\mathbf{WT}}$ and $\widetilde{\mathbf{WS}}$ are the corresponding stochastic matrices after normalizing **WW**, **WT** and **WS**.

Normalization: As $\mathbf{P}_T$ and $\mathbf{P}_S$ are known and fixed, we transform the w-th row vector $\mathbf{P}_{w.}^{(k+1)}(w \in W)$ to emotion distribution $\widetilde{\mathbf{P}}_{w.}^{(k+1)}$ by column-wise and row-wise min-max normalization in each round as follow:

$$\widetilde{\mathbf{P}}_{w.}^{(k+1)} = <\frac{\mathbf{P}_{w,1}^{(k+1)} - min_1}{(max_1 - min_1) \times Z}, ..., \frac{\mathbf{P}_{w,f}^{(k+1)} - min_f}{(max_f - min_f) \times Z}>$$

where notations $min_i = \min\{\mathbf{P}_{w,i}^{(k+1)}|w \in W\}$, $max_i = \max\{\mathbf{P}_{w,i}^{(k+1)}|w \in W\}$ and $Z = \sum_{i=1}^f \frac{\mathbf{P}_{w,i}^{(k+1)} - min_i}{(max_i - min_i)}$. This ensures that the values in the i-th column vector $\widetilde{\mathbf{P}}_{.i}^{(k+1)}(1 \leq$

286

Algorithm 1: Multi-Label Random Walk Algorithm

Input : word-word matrix $\widetilde{\mathbf{WW}}$; word-emoticon matrix $\widetilde{\mathbf{WT}}$; word-seed word matrix $\widetilde{\mathbf{WS}}$; word output matrix $\mathbf{P}_W$; emoticon output matrix $\mathbf{P}_T$; seed word output matrix $\mathbf{P}_S$; relative weight $\beta = \{\beta_1, \beta_2\}$; damping factor α; threshold ε.

Output: word output matrix $\mathbf{P}_W$

1 **foreach** *emotion word w of word set W* **do**
2 $\quad$ $\mathbf{P}_w$ is initialized with a uniform distribution
3 **foreach** *emoticon t of emoticon set T* **do**
4 $\quad$ $\mathbf{P}_t$ is initialized with v_t
5 **foreach** *word s of seed emotion word set S* **do**
6 $\quad$ $\mathbf{P}_s$ is initialized with v_s
7 $k = 1$
8 **repeat**
9 $\quad$ $\mathbf{P}_W^{(k+1)} = (1 - \alpha)\mathbf{e} + \alpha\mathbf{X}_W$
10 $\quad$ **foreach** *candidate word w of word set W* **do**
11 $\quad\quad$ $\mathbf{P}_{w.}^{(k+1)}$ is normalized to emotion vector $\widetilde{\mathbf{P}}_{w.}^{(k+1)}$
12 $\quad$ $\mathbf{P}_W^{(k+1)} = \widetilde{\mathbf{P}}_W^{(k+1)}$
13 $\quad$ $k = k + 1$
14 **until** $\| \boldsymbol{P}_W^{(k+1)} - \boldsymbol{P}_W^{(k)} \| < \varepsilon$
15 **return** $\boldsymbol{P}_W = \boldsymbol{P}_W^{(k+1)}$

$i \leq f$) are within $[0, 1]$ and comparable among different columns, and Z makes the sum of all elements in the vector equal to 1. The detail of MLRW algorithm is described in Algorithm 1.

We obtain the vectors of emotion distribution for all candidate words in W based on Algorithm 1. We reserve the candidate words whose probability of the main emotion in the resulted distribution $\max\{P_i | 1 \leq i \leq f\}$ larger than assigned threshold τ are selected into final emotion lexicon. We show some example entries of our generated emotion lexicon based on Chinese Sina Weibo in Table 2.

In Table 2, we show some example entries of our generated lexicon, which include typical emotion words or popular buzzwords in social networks. The scores in lexicon represents the strength of a given word with respect to the corresponding dimensions we consider. For example, 作呕 (feel like vomiting) has a predominant weight in *disgust* (0.35), 笑哈哈 (laughter) has a predominant weight in *happiness* (0.44), while 孬种(coward) has predominant weights in *digust* and *anger* (0.30 and 0.30 respectively). Therefore, the weights can reflect multiple complex emotion of the word. In addition, we also display some network buzzwords with superscript $\star$. Network buzzword conveys strong personal emotion. For example, 屌丝 (loser), 尼玛 (damn), 抽风 (puzzling behavior), 童鞋 (classmate) and 给力 (brilliant/awesome). As we can see, the lexicon captures emotion distribution of network buzzwords in real-world scenarios well. The lexicon generated contains 17k entries in total.

6. EXPERIMENTAL EVALUATION

We conduct experiments based on real-world Chinese microblogs from Sina Weibo[6] (a popular Twitter-like online so-

cial network in China) since Chinese emotion lexicons, especially with fine-grained emotion categories, are not common. However, our method can be generalized to other languages easily.

6.1 Data Resources

We crawled 3.5 million public Chinese microblogs with emoticons, via Weibo API[7], to construct our lexicon. The microblog posts are crawled randomly so that they are not limited to particular topics. By covering any possible topics, the dataset is expected to produce a more general emotion lexicon. We preprocess the dataset to obtain a cleaner corpus. The microblogs without typical emoticons (See Section 3.2) or selected seed words (See Table 7) are discarded; the ones containing negation words (e.g., 不 (not)) or adversative conjunctions(e.g., 但是 (but) and 然而 (however)) are filtered because they will complicate the inference of emotion orientation. Finally, we obtain a large enough corpus with 1.5 million microblogs.

Different from English, there is no separator between adjacent Chinese words. Therefore, we utilize NLPIR[8], a state-of-the-art Chinese word segmenter, to segment microblogs into words and assign them Part-of-Speech (POS). In order to guide the word segmenter to identify emotional network buzzwords more accurately, we introduce a self-built network buzzword lexicon named NetLex with these network buzzwords[9]. In addition, we use existing lexicons (i.e., sentiment lexicons HowNet[10], NTUSD [15] and emotion lexicon EWN) to filter unemotional words. Note that, we resort to these lexicons only to identify candidate words but neglect their original emotion labels in lexicons. Basically, we adopt the following two rules to select candidate words from the corpus: (1) Meaningless stop words and infrequent words appearing less than 10 times are discarded; (2) Words existing in aforementioned lexicons and POS tagged words (adjective, verb, noun and adverb)[11] are reserved. Finally, we consider all the remaining words as candidate words and construct our fine-grained emotion lexicon based on these words. Our constructed emotion lexicon has been made publicly available[12].

Recall that we quantitatively evaluate the constructed lexicon through an emotion classification task. For this task, we use an annotated Chinese microblog dataset named EACWT (i.e., Emotion Analysis in Chinese Weibo Texts) to build the training set for inferring the emotion distribution of emoticons and the test set for evaluating lexicon-based emotion classification. EACWT dataset is the standard corpus used in NLPCC2014 emotion analysis shared task[13]. Each sentence in EACWT is annotated with one or two emotion labels that represent the major or the major and the secondary emotions, respectively. The task, namely Emotion Sentence Identification and Classification, aims to identify the emotion-bearing sentences and determine the emotion category for each sentence. We extract all sentences

---

[6] http://www.weibo.com/

[7] http://open.weibo.com/

[8] http://ictclas.nlpir.org/newsdownloads?DocId=389

[9] http://wangci.net/

[10] http://www.keenage.com/html/e_index.html

[11] Only part of most relevant subtypes of POS are considered.

[12] https://github.com/songkaisong/EmotionLexicon

[13] http://tcci.ccf.org.cn/conference/2014/dldoc/evsam1.rar (Sample data is only available on the website, but complete data is provided with our lexicon together.)

Table 2: Examples of entries in our emotion lexicon, where superscript ⋆ denotes the network buzzwords

Entries	disgust	sadness	anger	happiness	like	fear	surprise
作呕 (feel like vomiting)	**0.35**	0.15	0.22	0.04	0.04	0.01	0.09
孬种 (coward)	**0.30**	0.10	**0.30**	0.03	0.06	0.16	0.05
嚎啕大哭 (burst into tears)	0.19	**0.33**	0.07	0.11	0.13	0.12	0.05
于心有愧 (have a guilty)	0.17	**0.40**	0.06	0.08	0.11	0.14	0.04
不共戴天(absolutely irreconcilable hatred)	0.34	0.09	**0.43**	0.01	0.00	0.08	0.05
怒火冲天 (towering rage)	0.32	0.09	**0.43**	0.00	0.01	0.08	0.07
笑哈哈 (laughter)	0.16	0.05	0.05	**0.44**	0.17	0.08	0.05
好笑 (funny)	0.22	0.06	0.07	**0.33**	0.14	0.09	0.09
好人 (good person)	0.07	0.05	0.04	0.17	**0.42**	0.22	0.03
甜美 (sweet)	0.16	0.06	0.03	0.20	**0.41**	0.08	0.06
鬼魔 (ghost)	0.09	0.04	0.04	0.15	0.22	**0.43**	0.03
地狱 (hell)	0.05	0.03	0.04	0.12	0.30	**0.44**	0.02
叹为观止 (amazing)	0.15	0.07	0.06	0.10	0.18	0.09	**0.35**
吃惊 (surprise)	0.16	0.11	0.09	0.15	0.11	0.12	**0.27**
童鞋 (classmate)⋆	0.17	0.07	0.06	0.26	**0.30**	0.07	0.07
屌丝 (loser)⋆	0.22	0.10	0.08	**0.26**	0.16	0.10	0.10
给力 (brilliant/awesome)⋆	0.15	0.04	0.04	**0.32**	0.30	0.07	0.08
抽风 (puzzling behavior)⋆	**0.24**	0.11	0.18	0.11	0.12	0.13	0.11
尼玛 (damn)⋆	**0.34**	0.12	0.30	0.04	0.04	0.10	0.06

with emotion labels, preprocess the sentences and then divide them into two parts: 3,600 sentences with emoticons are used for inferring emotion distribution of emoticons (see Section 3.1) and the remaining 6,799 labeled sentences are used for evaluating our learned lexicon.

Our method is implemented using the Java programming language based on a linear algebra package JAMA[14] for its efficiency. We run the program on a commodity PC with Intel Core i7-3537U CPU, 4G RAM and Windows-8 64-bit operating system.

6.2 Experiments and Results

We compare our lexicon with others built by the state-of-the-art methods. Rather than directly comparing the lexicons that may have different emotion types, which is difficult, we compare these automatically built emotion lexicons with respect to a manually created emotion lexicon called EWN [30]. EWN is regarded as the de facto standard because it has large size and the systems based on it has achieved state-of-the-art performance in sentiment classification evaluation [27]. We configure all methods compared in our experiments to produce word entries with the same seven possible emotions as EWN. We conduct two sets of experiments to examine (1) the quality of the lexicons generated by different methods; (2) the performance of sentence-level emotion classification based on these different lexicons.

6.2.1 Quality of Generated Lexicons

Evaluation Metrics: We use Precision, Recall and F-measure to assess the quality of the lexicons generated by different methods with respect to the manual lexicon EWN. We assess lexicon quality based on the major emotion because most of the EWN entry words are not provided with secondary emotion. We define the three metrics as follows:

$$P = \frac{\sum_{e \in E} |W_{EWN}(e) \cap W_{LEX}(e)|}{\sum_{e \in E} |W_{LEX}(e)|}$$

$$R = \frac{\sum_{e \in E} |W_{EWN}(e) \cap W_{LEX}(e)|}{\sum_{e \in E} |W_{EWN}(e)|}$$

where E is the set of all seven possible emotions, $W_{EWN}(e)$ is the word set with emotion label of e in EWN, $W_{LEX}(e)$ is the word set with emotion label e in the produced lexicon. And we further define F-measure as $F = \frac{2 \cdot P \cdot R}{P + R}$. High precision P indicates the generated lexicons performs well in capturing major emotion, and high recall R indicates a high coverage with respect to EWN.

In addition, we use an auxiliary metric KL-divergence [16] to assess whether generated lexicons have good word distribution over emotion classes. Similarly, we also define this metric based on the major emotion as follow:

$$D_{KL} = \sum_{e \in E} \log \left(\frac{P_{LEX}(e)}{Q_{EWN}(e)} \right) P_{LEX}(e)$$

where $P_{LEX}(e) = \frac{|W_{LEX}(e)|}{\sum_{e \in E} |W_{LEX}(e)|}$ is the proportion of words with emotion label e in EWN, $Q_{EWN}(e) = \frac{|W_{EWN}(e)|}{\sum_{e \in E} |W_{EWN}(e)|}$ is the proportion of words with emotion label of e in the produced lexicon. A better lexicon will have a smaller difference with EWN.

Baseline Methods: We compare our method with the following emotion lexicon generation approaches:

- **PMI_e:** Yang et al. [31] use a variant of PMI to get the correlation strength $co(w, t)$ between word w and emoticon t. Emoticons (see Figure 1) are classified into seven emotion sets, so emoticons in emotion set T_l are associated with the same emotion label l (i.e., $l \in \{1, 2, 3, 4, 5, 6, 7\}$). For each word w, its score under emotion class l is represented by maximum correlation strength based on emoticons in emotion set T_l. Then, w's emotion vector can be represented as $< \max_{t \in T_1} \{co(w, t)\}, ..., \max_{t \in T_7} \{co(w, t)\} >$.

- **PMI_s:** Xu et al. [30] use standard PMI to get the correlation strength $co(w, s)$ between word w and seed word s. Seed words and their emotion partition will

Table 3: Quality of lexicon on all emotion

	DM	PMI_s	PMI_e	Lex_s	Lex_e	Lex_c
P	0.202	0.282	0.361	0.403	0.484	**0.541**
R	0.060	0.083	0.106	0.118	0.143	**0.159**
F	0.092	0.128	0.164	0.183	0.221	**0.246**

Table 4: Statistics of idioms and non-idioms in EWN and Lex_c lexicons

	# of non-idioms	# of idioms
EWN	12,480	14,986
Lex_c	15,328	1,682

Table 5: Capacity of different lexicons

Lexicon	Emotion type	# of entries
HowNet	positive, negative	8,936
NTUSD	positive, negative	11,086
NetLex	no classification	675
Lex_c	seven emotions	17,010
EWN/Idiom	seven emotions	12,480
EWN	seven emotions	27,466

be introduced later (see Table 7). Similar to PMI_e, word w's score under emotion class l is represented by maximum correlation strength based on seed words in emotion set S_l. Then, the emotion of w can be represented as

$$< \max_{s \in S_1}\{co(w,s)\}, ..., \max_{s \in S_7}\{co(w,s)\} >.$$

- **DM**: Depeche Mood (DM) is an emotion lexicon proposed by Staiano and Guerini [25]. It first constructs the document-by-emotion matrix and the word-by-document matrix based on tf*idf, then applies matrix multiplication to represent each word as an emotion vector.

- **Lex_e:** This lexicon is generated using the special case of our model which only involves emoticons and candidate words in the graph. This essentially reduces to the model in [6] using the multiple emotion representation.

- **Lex_s**: This lexicon is generated as a special case of our model using only seed emotion words to produce emotion distributions of candidate words.

- **Lex_c**: This lexicon is built based on the full configuration of our proposed method that combines seed words and emoticons.

Results: As shown in Table 3, we find that the constructed lexicons Lex_c, Lex_e and Lex_s have obvious advantages over DM, PMI_e and PMI_s methods. Especially, the Lex_c combining the effects of seed words and emoticons performs the best, which manifests that our method can produce a high-quality and high-coverage emotion lexicon.

The following example illustrates such effect:
(1) My kitty **smirks** and sticks tongue out! 😛
(2) The star's **smirk** let me feel *disgusting*.
In (1), the candidate word 'smirk' co-occurring with the emoticon 😛 is assigned an emotion vector <0.16, 0.07, 0.04, **0.26**, 0.21, 0.23, 0.03> by Lex_e; In (2), 'smirk' co-occurring with the seed word 'disgusting' has an emotion vector <**0.40**, 0.15, 0.11, 0.04, 0.08, 0.12, 0.10> by Lex_s. We can see that in the respective lexicon, 'smirk' mainly conveys either 'happiness' or 'disgust', which is not complete. By considering both seed words and emoticons, Lex_c assigned 'smirk' the emotion vector <**0.24**, 0.09, 0.06, **0.24**, 0.13, 0.10, 0.14>, where the emotions *disgust* and *happiness* receive balanced weights, which is closer to its real emotion distribution.

We observe that the overlap with the manual lexicon EWN in Table 3 is notably low. This is because EWN contains too many idioms which are not commonly used in the microblog. We show statistical results of idioms and non-idioms in Lex_c and EWN in Table 4, respectively. From Table 4, we find that idioms and non-idioms in EWN show almost the same proportions, but Lex_c has a much larger percentage of the non-idioms. Therefore, too many idioms in EWN which are rarely used in microblog posts leads to a low recall in

Table 3. But our Lex_c lexicon built from the corpus has much more non-idioms and can be extended automatically with the increase of corpus size.

We further compare the capacity of Lex_c with existing binary emotion lexicons HowNet and NTUSD, fine-grained emotion lexicon EWN and its simplified version exclusive of idioms EWN/Idiom, and network buzzword lexicon NetLex in Table 5. Although these available lexicons with different emotion types are based on different construction methods, we still compare them together for indicating that our generated lexicon is large enough to be used. From Table 5, we can easily find that Lex_c is much larger than most available Chinese lexicons HowNet, NTUSD, NetLex and EWN/Idiom in size, which indicates that Lex_c is more universal compared to most existing lexicons.

As most candidate words are supposed to be emotional based on specific context, we reserve as many words as possible. As a consequence, the lexicons built automatically by our method and other comparable methods are similar in size. However, the distributions of these words in each lexicon under each emotion class are remarkably different and displayed based on the major emotion in Table 6. It is difficult to tell whether Lex_c has a good word distribution from Table 6. Therefore, we again resort to auxiliary metric KL-divergence to measure the difference between each generated lexicon with respect to the manual lexicon EWN. The results are plotted in Figure 3. We can easily find that our Lex_c performs the best by achieving the smallest KL-divergence between Lex_c and EWN. In contrast, DM performs the worst in identifying words under emotion *like*, and PMI_e behaves poor since it identifies few words under emotion *disgust* and *fear*.

We follow the semi-automatic approach (see Section 4) and select 5 representative seed words respectively from each

Table 6: Word distributions of different lexicons, where di, sa, an, ha, li, fe, su denote the seven emotions *disgust*, *sadness*, *anger*, *happiness*, *like*, *fear* and *surprise*, respectively

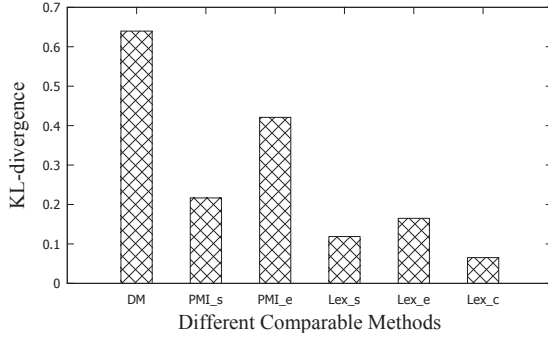|	PMI_s	PMI_e	DM	Lex_s	Lex_e	Lex_c	EWN
di	7,805	394	4,105	6,909	4,921	5,822	10,282
sa	1,627	3,384	1,568	1,224	22	401	2,314
an	1,307	2,433	2,986	1,063	63	642	388
ha	1,862	2,909	2,297	1,930	3,367	3,034	1,967
li	2,049	7,495	848	2,944	6,273	6,205	11,108
fe	1,389	16	2,102	1,111	2,716	670	1,179
su	1,382	825	3,173	840	34	236	228

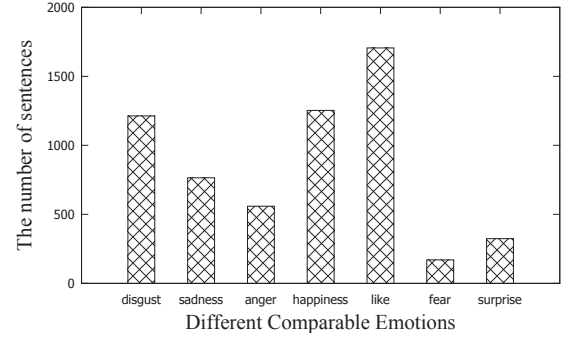**Figure 3: Word distribution under each emotion**

Figure 4: The number of testing sentences under each emotion category

Table 7: Seed Emotional Words

Emotion	Seed Words
disgust	讨厌 (disgusting), 多心 (suspicious) 厌烦 (boredom), 嫉妒 (jealousy), 可耻 (shame)
sadness	忧伤 (distressed), 伤心 (heart-broken) 绝望 (despair), 内疚 (compunction), 悲伤 (sad)
anger	恼火 (annoyed), 生气 (angry), 气愤 (furious) 悲愤 (grief and indignation), 火气 (anger)
happiness	高兴 (happy), 喜悦 (delightful), 欢喜 (joyful) 安心 (relieved), 踏实 (sureness)
like	尊敬 (respect), 赞扬 (praise), 相信 (believe) 喜爱 (inlove), 祝愿 (wish)
fear	害怕 (afraid), 恐惧 (dread), 吓人 (scary) 可怕 (horrible), 畏惧 (fear)
surprise	惊呆 (stunned), 震惊 (shock), 惊讶 (amazed) 惊奇 (surprise), 惊人 (astonishing)

emotion class. The 35 seed words from seven emotions we considered are shown in Table 7.

6.2.2 Sentence-level Emotion Classification

We also quantitatively evaluated the learned emotion lexicons through a sentence-level emotion classification task. The objective of the task is to assign the major emotion and secondary emotion to each sentence in test dataset from the NLPCC2014 corpus. We used a simple voting-based algorithm to assign class labels to a given sentence as follows. Let an entry word w in the lexicon represented by emotion distribution $< P_1^w, ..., P_7^w >$. For each emotion, we add up the values of all the emotion words contained in the sentence by looking up the lexicon:

$$sen \rightarrow < \sum_{i=1}^{n} P_1^{w_i}, ..., \sum_{i=1}^{n} P_7^{w_i} >$$

where n is the number of emotion words in sentence *sen*. Note that, this paper is not about improving the sentiment classification; we rather use emotion classification task as the standard task for measuring and comparing different lexicon generation methods. Therefore, we do not compare with emotion classification methods which are not based on lexicons.

Evaluation Metrics: We use popular Macro Metric [34] and Average Precision [33] to measure the effectiveness of single-label and multi-label emotion classification, respectively, based on the produced lexicons.

- Macro Metric: Macro Precision (MaP), Macro Recall (MaR) and Macro F-measure (MaF):

$$MaP = \frac{1}{7} \sum_i \frac{\#correct(emo_i)}{\#label(emo_i)}, MaR = \frac{1}{7} \sum_i \frac{\#correct(emo_i)}{\#gold(emo_i)}$$

where emo_i is any type of emotion, $\#correct(emo_i)$ is the number of microblogs with emo_i recognized correctly by algorithm, $\#label(emo_i)$ is the number of microblogs with emo_i recognized by algorithm and $\#gold(emo_i)$ is the number of microblogs labeled with emo_i. $MaF = \frac{2 \cdot MaP \cdot MaR}{MaP + MaR}$.

- Average Precision: For the top-2 emotion classification, we use Average Precision (AP) in multi-label classification as metric:

$$AP = \frac{1}{|n|} \sum_{j=1}^{n} \frac{1}{|Y_j|} \sum_{k=1}^{|Y_j|} \frac{|(emo \in Y_j | r(x_j, emo) \leq r(x_j, emo_k))|}{r(x_j, emo_k)}$$

where n is the number of sentences in test set, Y_j is the set of standard emotion labels for sentence x_j, *emo* is the system predicted emotion label, emo_k is one of the ground-truth emotions of x_j, and $r(x_j, y)$ is the ranked position of emotion y in x_j. Note that major emotion is always put ahead of secondary emotion, which gives an order of emotion labels in both ground-truth and prediction. AP indicates the average fraction of relevant labels ranked higher than the true label.

Parameter Setting: There are three adjustable parameters in our method, including the damping factor α, and the relative weights β_1 and β_2. According to the suggested setting to PageRank-like algorithm [2, 21], we set $\alpha = 0.85$. For tuning β_1 and β_2, we randomly choose 799 sentences into a development set, and the remaining 6,000 sentences are used for test. We display the number of testing sentences under each emotion category in Figure 4, which also reflects the fact that *like*, *happiness* and *disgust* play major roles in emotion expression. We use the parameter setting that gives the optimal F-measure value. For Lex_c, the optimal settings is $\beta_1 = 0.5$ and $\beta_2 = 0.4$; Lex_e and Lex_s just have a single β. We set it as 0.4 and 0.45, respectively. We set the convergence threshold $\varepsilon = 1e - 5$ and major emotion threshold $\tau = 0.20$ empirically.

Results: The results for single-label classification are shown in Table 8. We notice that the Lex_c model outperforms all the other automatic lexicon building methods. Its F-measure is just slightly lower than that of the manual lexicon EWN by 6.47%. This again confirms that our method

Table 8: Single-label emotion classification

Methods	MaP	MaR	MaF
PML_s	0.258	0.283	0.270
PML_e	0.390	0.313	0.347
DM	0.348	0.368	0.357
Lex_s	0.318	0.293	0.305
Lex_e	0.418	0.273	0.330
Lex_c	**0.579**	0.328	0.419
EWN	0.494	**0.411**	**0.448**

Table 9: Comparison of average occurrence number of seed words and emoticons in corpus

	Average occurrence number
Seed word	1,155
Emoticon	55,637

Figure 5: Multi-label emotion classification

combining emoticons and seed words is more effective in capturing major emotion of emotional words. We also find that the Lex_s and PML_s models do not perform as well as the DM, PML_e and Lex_e methods in terms of F-measure. This is mainly because seed words appear much fewer times than emoticons in the corpus. We further calculate the average occurrence number of seed words and emoticons, respectively, in corpus by formulas $\frac{\#emoticons}{\#microblogs}$ and $\frac{\#seedwords}{\#microblogs}$, and display results in Table 9 which explains the reasons for lower performance of Lex_s and PML_s models intuitively. The performance of Lex_e is close to PML_e and DM since all these methods are based on emoticons.

Then, we further study the multi-label emotion classification performance. The results are provided in Figure 5. We observe that PML_s performs much worse than the other models by failing to capture the secondary emotion. This is because PML_s does not leverage emoticons, which is especially a disadvantage for this multi-label task due to its potential incomplete emotion representation. Lex_c performs nearly as well as the manually built lexicon EWN, but Lex_e and Lex_s performs much worse, which implies that both emoticons and seed words contribute to estimating emotion distribution of words more accurately.

7. CONCLUSION AND FUTURE WORK

In this paper, we focus on building a high-quality emotion lexicon automatically from massive collection of microblogs. Our idea is to capture the emotion distributions of candidate words that convey multiple complex emotions by combining the effect of seed words and emoticons that co-occur with the candidate words. We resort to a three-layer heterogeneous graph to represent emoticons, seed words and candidate words and the correlations among them, on which a multi-label random walk algorithm is performed to strengthen the estimation of emotion distributions of candidate words. Experimental results based on real-world microblogs demonstrate that the performance of our lexicon in capturing words' emotion is nearly as well as a high-quality emotion lexicon created manually. Meanwhile, it outperforms other lexicons created by the state-of-the-art automatic methods in emotion classification.

In the future, we will introduce the syntax unit composed of word and Part-of-Speech into emotion lexicon building and study the emotion distribution of emotional words un-
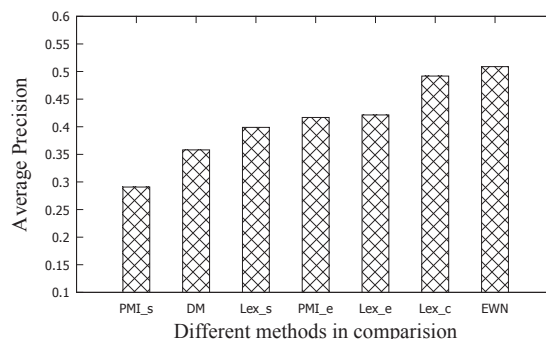
der each POS. In addition, we will further improve the quality and capacity of our Chinese emotion lexicon Lex_c, and publish lexicons in other languages.

8. ACKNOWLEDGMENTS

9. REFERENCES

[1] P. P. Alexander Pak. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 1320–1326, 2010.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):101–117, 1998.

[3] E. Cambria, R. Speer, C. Havasi, and A. Hussain. enticnet: A publicly available semantic resource for opinion mining. In *Proceedings of AAAI Fall Symposium on Commonsense Knowledge*, pages 417–422, 2010.

[4] Y. Chen and S. Skiena. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 383–389, 2014.

[5] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 417–422, 2006.

[6] S. Feng, K. Song, D. Wang, and G. Yu. A word-emoticon mutual reinforcement ranking model for building sentiment lexicon from massive collection of microblogs. *World Wide Web*, 18(4):949–967, 2015.

[7] S. Feng, D. Wang, G. Yu, W. Gao, and K.-F. Wong. Extracting common emotions from blogs based on fine-grained sentiment clustering. *Knowledge and Information Systems*, 27(2):281–302, 2011.

[8] D. Gao, F. Wei, W. Li, X. Liu, and M. Zhou. Co-training based bilingual sentiment lexicon learning. In *Proceedings of AAAI Workshops on Late-Breaking Developments in the Field of Artificial Intelligence*, 2013.

[9] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. In *Technical report, Stanford*, 2009.

[10] Y. He, C. Lin, W. Gao, and K.-F. Wong. Tracking sentiment and topic dynamics from social media. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, pages 483–486, 2012.

[11] Y. He, C. Lin, W. Gao, and K.-F. Wong. Dynamic joint sentiment-topic model. *ACM Transactions on Intelligent Systems and Technology*, 5(1):6:1–6:21, 2013.

[12] C. J. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the 8th International Conference on Weblogs and Social Media*, 2014.

[13] N. Kaji and M. Kitsuregawa. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of EMNLP-CoNLL*, pages 1075–1083, 2007.

[14] R. Krestel and S. Siersdorfer. Generating contextualized sentiment lexica based on latent topics and user ratings. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 129–138, 2013.

[15] L.-W. Ku and H.-H. Chen. Mining opinions from the web: Beyond relevance retrieval. *Journal of American Society for Information Science and Technology*, 58(12):1838–1850, 2007.

[16] S. Kullback and R. Leibler. On information and sufficiency. *Annals of MathematicalStatistics*, 22:79–87, 1951.

[17] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[18] Y. Lu, M. Castellanos, U. Daya, and C. Zha. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, pages 347–356, 2011.

[19] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[20] S. Mohammad and P. D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[21] L. Page, S. Brin, R. Motwami, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Technical Report 1999-0120, Computer Science Department, Standford University*, 1999.

[22] F. Peleja, J. Santos, and J. Magalhaes. Reputation analysis with a ranked sentiment-lexicon. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1207–1210, 2014.

[23] Y. Rao, J. Lei, W. Liu, Q. Li, and M. Chen. Building emotional dictionary for sentiment analysis of online news. *World Wide Web*, 17(4):732–742, 2014.

[24] K. Song, S. Feng, W. Gao, D. Wang, G. Yu, and K.-F. Wong. Personalized sentiment classification based on latent individuality of microblog users. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015.

[25] J. Staiano and M. Guerini. Depeche mood: a lexicon for emotion analysis from crowd annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 427–433, 2014.

[26] C. Strapparava and A. Valitutti. Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004.

[27] S. Wen and X. Wan. Emotion classification in microblog texts using class sequential rules. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 187–193, 2014.

[28] G. Xu, X. Meng, and H. Wang. Build chinese emotion lexicons using a graph-based algorithm and multiple resources. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1209–1231, 2010.

[29] J. Xu, R. Xu, Y. Zheng, Q. Lu, K.-F. Wong, and X. Wang. Chinese emotion lexicon developing via multi-lingual lexical resources integration. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 2*, pages 174–182, 2013.

[30] L. Xu, H. Lin, Y. Pan, H. Ren, and J. Chen. Constructing the afective lexicon ontology. *Journal of The China Society For Scientific and Technical Information*, 27(2):180–185, 2008.

[31] C. Yang, K. H.-Y. Lin, and H.-H. Chen. Building emotion lexicon from weblog corpora. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 133–136, 2007.

[32] M. Yang, D. Zhu, and K.-P. Chow. A topic model for building fine-grained domain-specific emotion lexicon. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 421–426, 2014.

[33] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

[34] Z. Zhang and M. P. Singh. Renew: A semi-supervised framework for generating domain-specific lexicons and sentiment analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 542–551, 2014.

[35] J. Zhao, L. Dong, J. Wu, and K. Xu. Moodlens: An emoticon-based sentiment analysis system for chinese tweets in weibo. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1528–1531, 2012.

[36] W. Zheng, C. Wang, Z. Liu, and J. Wang. A multi-label classification algorithm based on random walk model. *Chinese Journal of Computers*, 33(8):1418–1426, 2010.