

## Research Article

# A Self-Adaptive Hidden Markov Model for Emotion Classification in Chinese Microblogs

Li Liu,<sup>1,2,3</sup> Dashi Luo,<sup>2</sup> Ming Liu,<sup>4</sup> Jun Zhong,<sup>2</sup> Ye Wei,<sup>2</sup> and Letian Sun<sup>2</sup>

<sup>1</sup>School of Software Engineering, Chongqing University, Chongqing 401331, China

<sup>2</sup>School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China

<sup>3</sup>School of Computing, National University of Singapore, Singapore 117417

<sup>4</sup>Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China

Correspondence should be addressed to Ming Liu; [mingliu@swu.edu.cn](mailto:mingliu@swu.edu.cn)

Received 24 September 2014; Accepted 18 December 2014

Academic Editor: Praveen Agarwal

Copyright © 2015 Li Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Microblogging is increasingly becoming one of the most popular online social media for people to express ideas and emotions. The amount of socially generated content from this medium is enormous. Text mining techniques have been intensively applied to discover the hidden knowledge and emotions from this huge dataset. In this paper, we propose a modified version of hidden Markov model (HMM) classifier, called self-adaptive HMM, whose parameters are optimized by Particle Swarm Optimization algorithms. Since manually labeling large-scale dataset is difficult, we also employ the entropy to decide whether a new unlabeled tweet shall be contained in the training dataset after being assigned an emotion using our HMM-based approach. In the experiment, we collected about 200,000 Chinese tweets from Sina Weibo. The results show that the  $F$ -score of our approach gets 76% on happiness and fear and 65% on anger, surprise, and sadness. In addition, the self-adaptive HMM classifier outperforms Naive Bayes and Support Vector Machine on recognition of happiness, anger, and sadness.

## 1. Introduction

In the recent years, online social media, such as Microblogging, has generated enormous content on the world wide web. Microblogs are extremely limited in the length of 140 characters. This allows users to easily update a Microblog or receive updates from mobile devices, such as cell phones. Twitter has grown to one of the most popular Microblogging websites and generated hundreds of millions updates per day. Similar to Twitter, Weibo is a Microblogging website in China. In the past few years, it has become an increasingly important source of online social media information with its population of Chinese users growing rapidly to 309 million in 2012 ([http://news.xinhuanet.com/english/sci/2013-01/15/c\\_132104473.htm](http://news.xinhuanet.com/english/sci/2013-01/15/c_132104473.htm)), as well as with more than 1,000 tweets generated in every second.

The emotional states of the users are able to be inferred from these large numbers of short tweets. Emotions in tweets play a significant role in many fields. The stock market and other socioeconomic phenomena could be predicted by using

emotion analysis of the Twitter users [1]. Even the gross happiness of a community or a country could be estimated from Twitter [2].

Text mining on Chinese tweets is a challenging work. First, word segmentation in Chinese is more difficult than in English since there is no space between Chinese characters and it requires disambiguating segmentation strings. Second, like English Twitter, new words are coming out every day and it is difficult to develop a system to recognize the unknown emotional words. Third, words are ambiguous in various contexts, especially those emotional words. Most of the recent sentiment analysis approaches [3, 4] for Chinese tweets employ emotional word count as main feature and build emotional word dictionary for inference.

Considering these issues, we propose a self-adaptive hidden Markov model (HMM) based method to perform the emotion analysis for Chinese tweets. Our method implements a self-adaptive mechanism, which learns the parameters of HMM models and appends new recognized emotional words or sentences into emotional word dictionary, to deal

with the issues of the sentiment ambiguities of words and the generation of new words. The main contributions of this paper are shown as follows. (a) More fine grained emotional categories are recognized. We employ the well-known six basic emotional categories defined by Ekman [5]: happiness, sadness, fear, anger, disgust, and surprise, which are more intuitive and useful than traditional sentiment analysis categories: positive, negative, and neutral categories. (b) More useful features, other than word count, are defined. Our method employs the category-based features extracted from the short sentences to train our proposed HMM models. (c) A self-adaptive mechanism is used to test on a real dataset collected from Sina Weibo.

The rest of the paper is organized as follows. In Section 2, we survey the related works of emotion analysis either in Twitter or in Weibo and provide background on the concepts and methods related to emotion analysis for tweets, especially for Chinese tweets. In Section 3, we describe our proposed self-adaptive HMM-based method. In Section 4, we illustrate the dataset and the experiment setup and present results from our study. We discuss limitations of the method and future work and conclude in Section 5.

## 2. Background

*2.1. Related Work.* Weibo plays a significant role in people's lives; therefore its opinion mining and emotion analysis become interesting researches. There are a lot of researches on Twitter emotions abroad. Spencer and Uchyigit [6] identified subjective tweets and detected their emotion polarity by using Naive Bayes. In their test using bigrams without part of speech (POS) tags has a high accuracy of 52.31%. In the study carried out by Pak and Paroubek [7], they utilized Naive Bayes to classify tweets into positive, negative, and neutral. It is shown that using the presence of an  $n$ -gram as a binary feature yielded the best results. A research [8] has shown that machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) had high accuracy rate with above 80% when sentiment categories consisted only of positive and negative. In the detection based on Twitter corpus, the authors explored various strategies of selecting features and found that bigram features outperform unigram and POS.

However, studies of Chinese Weibo emotions grow just a few years. Zhao et al. [9] used Naive Bayes model to train on emoticon features; they classified Chinese tweets to four categories of emotions (i.e., angry, disgusting, joyful, and sad), with an empirical precision of 64.3%. Yuan and Purver [10] classified Chinese tweets to six emotions which are defined by Ekman as well. In their experiments, character-based features achieved 80% accuracies for "happiness" and "fear" based on the SVM, but there is insufficient analysis to the other four emotions.

Our self-adaptive HMM is to recognize more grained emotional categories by category-based features. Besides, the self-adaptive mechanism can continually enhance the recognition accuracy.

*2.2. Emotion Models.* Two models are normally used to represent emotions: the categorical model and dimensional model [11]. The categorical models are based on the assumption that the emotional categories are distinct. Ekman defined six basic emotions: anger, disgust, fear, joy, sadness, and surprise, and found a high agreement of expressions in multiple culture groups in his study. D'Mello et al. [12] proposed five categories (boredom, confusion, delight, flow, and frustration) for describing the affect states in ITS interactions.

In the dimensional model, core affects represent emotions in a two- or three-dimensional space. A valence dimension represents positive and negative emotions on different ends of scale. The arousal dimension distinguishes excited states from calm states. Sometimes a third, dominance dimension is used to differentiate if the subject feels in control of the situation or not. For example, Positive Affect and Negative Affect Schedule (PANAS) [13] provides two opposite mood factors (positive and negative) which has been widely used for opinion mining.

In this study, the reason we employ Ekman's emotion model is that the six basic emotions are distinct expressions and contain people daily mood.

*2.3. Chinese Text Preprocessing.* Chinese, a unique language, differing from another language like English, is written without word delimiters. Therefore, its word segmentation is a significant task. Chinese text preprocessing is divided into two steps: word segmentation and stop words removing.

To split words, there are some Chinese lexical analysis systems (Institute of Computing Technology, Chinese Lexical Analysis System (ICTCLAS) [14], IKAnalyzer (<http://code.google.com/p/ik-analyzer/>), HTTPCWS (<http://code.google.com/p/httpcws/>), Simple Chinese Word Segmentation (SCWS) (<http://gc.codehum.com/p/libscws/>), which are developed to perform word segmentation task). The accuracy rates of those systems are all evaluated above 90 percent.

We utilized NLPPIR (namely, ICTCLAS 2013) as a word splitter in this study. Its latest lexical database contains new words often appearing in Weibo. It also can recognize bloggers' nickname. Furthermore, an adaptive word splitter can add new words in its lexical database, which can be applied on our self-adaptive mechanism to update word-emotional vocabulary.

For reducing dimension of feature space, the next important step is to remove stop words (quantifier, pronoun, digits, notations, etc., e.g., "hundred," "we," "3," and "%"). Similarly, Weibo has many unique properties like usernames and usage of links that must be removed.

*2.4. Features.* In the field of document classification, each document is represented as a vector of term. Effective feature extraction is essential to make learning task effective. We adopt four features for our emotion classification model as follows.

*Mutual Information (MI).* MI is a useful information measure, which refers to the correlation between two sets of

events. MI reflects the relevance between terms and text categorization on emotions. Two events mutual information of term  $t$  and emotion  $c_i$  is defined as

$$\text{MI}(t, c_i) = \log \frac{P(t | c_i)}{P(t)}. \quad (1)$$

*Chi-Square (CHI)*. CHI is a statistical method to measure the lack of independence between term  $t$  and emotion  $c_i$ . The higher the CHI values, the more dependence between them:

$$\chi(t, c_i) = \frac{(AD - BC)^2 \times N}{(A + B)(C + D)(A + C)(B + D)}, \quad (2)$$

where  $A$  represents presence of  $t$  and membership in  $c_i$ ;  $B$  represents presence of  $t$  and nonmembership in  $c_i$ ;  $C$  represents absence of  $t$  and membership in  $c_i$ ;  $D$  represents absence of  $t$  and nonmembership in  $c_i$ ;  $N$  is the total number of tweets.

*Term Frequency-Inverse Document Frequency (TF-IDF)*. The main idea of TF-IDF is as follows: if a word or phrase in a tweet appears in a high frequency and rarely appears in other tweets; then it has a good ability to distinguish among emotions:

$$\text{TF-IDF}(t, c_i) = \text{TF}_{i,t} \times \text{IDF}_t, \quad (3)$$

where  $\text{TF}_{i,t}$  represents the frequency of the term  $t$  in the emotion  $c_i$ ; the main idea of  $\text{IDF}_t$  is that if the term  $t$  rarely appears in other emotions,  $\text{IDF}_t$  can be an indicator feature for  $c_i$ . Their definition is

$$\text{TF}_{i,t} = \frac{N_{i,t}}{\sum_k N_{k,t}}; \quad \text{IDF}_t = \log \left( \frac{N}{n_t} + 0.01 \right), \quad (4)$$

where  $N_{i,t}$  means the number of how many times  $t$  occurs in emotion  $c_i$ ,  $N$  denotes the total number of tweets in the dataset, and  $n_t$  is the number of tweets where  $t$  appears.

*Expected Cross Entropy (ECE)*. ECE reflects the probability distribution of the text category on emotions and distances between the probability distribution given term  $t$ . Its definition is

$$\text{ECE}(t, c_i) = P(c_i | t) \log \frac{P(c_i | t)}{P(c_i)}, \quad (5)$$

where  $P(c_i | t)$  represents the probability of emotion  $c_i$  given term  $t$  and  $P(c_i)$  represents the probability of tweets associated with emotion  $c_i$ .

## 2.5. Classification Methods

*Hidden Markov Model (HMM)*. HMM is used to describe a Markov process with unknown parameters. It is difficult to determine implicit parameters of the process through observable parameters, which are then used to make further analysis.

An HMM describes two related discrete-time stochastic processes. The first process pertains to hidden state variables, denoting  $(S_1, S_2, \dots, S_u)$ , which emit observed variables with different probabilities, and the second process pertains to observed variables  $(y_1, y_2, \dots, y_u)$ .

Main parameters of HMM are the transition and emission probabilities [15]:

$P(S_k = s_p | S_{k-1} = s_q)$  (transition probabilities): it means current state is depend on the previous state  $s_q$ ;

$P(y_k | S_k = s_p)$  (emission probabilities): observation symbol is released by current state  $s_p$ .

In our model, an HMM model  $\lambda^{(c_i)}$  is built for each emotion  $c_i$ . The features extracted from a tweet  $T$  are observed variables, while each hidden state  $S_k^{(c_i)}$  of  $\lambda^{(c_i)}$  is considered as a state associated with the feature  $y_k$ . If the tweet  $T$  gets the highest probability on the model  $\lambda^{(c_i)}$ , it means that  $T$  is associated with emotion  $c_i$ .

## 3. Method

*3.1. Features as Observed Variables*. Feature extraction methods fall into two main categories: category-based extraction and global-based extraction. The features exacted by the latter methods can reflect the importance of words in global corpus but cannot be used to distinguish the differences between emotions. Therefore, we adopt category-based feature extraction methods in our models. All the selected features we introduced in Section 2.4 are category-based extracted features.

Suppose a set of  $u$  features extraction methods  $\{F_1, F_2, \dots, F_u\}$ . A tweet  $T$  is first divided into  $n$  terms  $(w_1, w_2, \dots, w_n)$ . Let  $x_{jk}$  be the  $k$ th feature of the word  $w_j$  extracted by the method  $F_k$ . Then, an intermediate  $n \times u$  matrix of term-level features is obtained for calculating the tweet-level feature vector. For each emotion  $c_i$  ( $1 \leq i \leq 6$ ), the tweet-level feature vector of the tweet  $T$  can be calculated as follows:

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \rightarrow \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1u} \\ x_{21} & x_{22} & \cdots & x_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nu} \end{bmatrix} \rightarrow [y_1 \ y_2 \ \cdots \ y_u]^{(c_i)}, \quad (6)$$

where  $y_k = \sum_{j=1}^n x_{jk}/n$ , ( $1 \leq k \leq u$ ). Actually,  $y_k$  is a mean value of  $k$ th features over all the words.

In our experiment, a tweet is mapped into a four-dimensional vector using the four features: MI, CHI, TF-IDF and ECE. For example, the tweet ‘‘How lovely!’’ is divided into two terms: ‘‘how’’ and ‘‘lovely’’ in Chinese. The features of the terms on happiness and anger are calculated, respectively, as listed in Table 1.

Therefore, the tweet’s feature vector is [0.0022, 0.0107, 0.0052, 0.0016] in happiness and [−2.8E−4, 3.9E−4, 1.1E−4, −3.2E−5] in anger.

**Input:** feature vector  $[y_1 \ y_2 \ \dots \ y_u]^{(c_i)}$   
**Output:** probability  $P([y_1 \ y_2 \ \dots \ y_u]^{(c_i)} | \lambda^{(c_i)})$   
**for**  $k = 1$  **to**  $u$  **do**  
    Calculate  $P(y_k | S_k)^{(c_i)}$   
**end for**  
Calculate  $P([y_1 \ y_2 \ \dots \ y_u]^{(c_i)} | \lambda^{(c_i)})$  using forward algorithm.

ALGORITHM 1:  $\lambda^{(c_i)}$  model algorithm.

TABLE 1: The features of the terms “how” and “lovely” on happiness and anger, respectively.

Term	Emotions	MI	CHI	TF-IDF	ECE
How	Happiness	0.0014	0.0147	0.00088	0.00059
	Anger	-0.00056	0.00063	0.00023	-0.000065
Lovely	Happiness	0.0030	0.0067	0.0095	0.0027
	Anger	0	0.00014	0	0

**3.2. HMM-Based Emotional Classification Model.** In our case each hidden state is supposed to emit a value, and the whole model generates the sequence of values that constitutes the tweet’s feature vector. States are considered to be a set of values that represent the best emotional category. There is a one-to-one mapping between HMM states and tweet features, which requires hidden states transition to be stationary and the states to begin at the first state  $s_1$ . When the classifier works, transition probabilities indicate features of test tweets being drawn closer to the emotion, which is given as follows:

$$P(S_k = s_p | S_{k-1} = s_q) = \begin{cases} 1, & (p = q + 1) \\ 0, & (p \neq q + 1). \end{cases} \quad (7)$$

The emission probability  $P(y_k | S_k)$  also can be calculated by the known state  $S_k$ , feature  $y_k$ , and training data. Our calculation method is given at the end of this chapter.

Algorithm 1 depicts the procedure of building an HMM model  $\lambda^{(c_i)}$  for emotion  $c_i$ . The tweet-level feature vector is denoted as  $[y_1 \ y_2 \ \dots \ y_u]^{(c_i)}$ .

We construct an HMM model for each of the six emotions. When a new tweet arrives, the probabilities of the tweet are calculated on the six models, respectively. The tweet is labeled with the emotion whose model is associated with the maximum probability.

**A Strategy of Calculating  $P(y_k | S_k)^{(c_i)}$ .** To find a way to calculate the  $P(y_k | S_k)$  of emotion  $c_i$ , denoted as  $P(y_k | S_k)^{(c_i)}$ , we attempt to use Jaccard similarity measure [16] to test correlation between value  $y_k$  and state  $S_k$ :

$$P(y_k | S_k)^{(c_i)} = J(y_k | S_k)^{(c_i)} = \frac{M_{11}}{M_{11} + M_{10} + M_{01}}, \quad (8)$$

where  $M_{11}$  represents the number of tweets, which contains both  $y_k$  and  $S_k$  in  $c_i$ ;  $M_{01}$  represents the number of tweets,

which contains only  $y_k$  in  $c_i$ ; and  $M_{10}$  represents the number of tweets, which contains only  $S_k$  in  $c_i$ . In order to check whether emotion  $c_i$  associates to observation  $y_k$  or state  $S_k$ , we introduce an associated factor, denoted by  $\delta_k$ . Let  $y'_k$  be the feature extracted from training data. The following describes the association:

$$|y'_k - y_k| \leq \delta_k \quad \text{or} \quad |y'_k - S_k| \leq \delta_k. \quad (9)$$

We assume that observation  $y_k$  or state  $S_k$  is associated with  $c_i$  if the above inequality is met.

### 3.3. Self-Adaptive Mechanism

**3.3.1. Parameters Computed by PSO.** To build an excellent HMM classifier, an important problem we have to solve is to find optimized sequences of HMM states. There are a variety of strategies we can utilize for optimizing state parameters.

Particle Swarm Optimization (PSO) algorithm [17] is a population based stochastic optimization. In our algorithm, each particle represents a candidate solution of an HMM parameter. These particles move around in the search-space. Through particle’s local best known position, it is guided toward the best known positions. Eventually this iterative method can find best parameters. Compared to Genetic Algorithm [18], PSO has simple rules and more powerful ability of global optimization that has been good applications in our study:

$$v[] = W \times v[] + c_1 \times r \times (pbest[] - present[]) \quad (a)$$

$$+ c_2 \times R \times (gbest[] - present[])$$

$$present[] = present[] + v[], \quad (b)$$

where  $[]$  indicates that its corresponding variable is a vector. The variables  $r$  and  $R$  are random numbers between  $[0, 1]$ .  $pbest[]$  records the individual extremes and  $gbest[]$  records the global extremes. The constant  $W$  is the inertia weight, and  $c_1$  and  $c_2$  present acceleration constants. Moreover, (a) updates velocities of particles according to previous velocity and the distance to the best particle. The equation (b) updates the particles’ position  $present[]$  according to its previous position and current velocity.

**PSO Parameter Settings.** Different PSO parameters may have a large impact on optimization performance. Following are guidelines to help us select PSO parameters [19, 20].



```

Input: training data pool  $L$ , test data pool  $U$ , query strategy  $\phi(\bullet)$ , query batch size  $B$ 
repeat
  for  $i = 1$  to  $|C|$  do
    Optimized  $\lambda^{(c_i)}$  by using current  $L$  and PSO algorithm;
  end for
  for  $b = 1$  to  $B$  do
     $T_b^* = \arg \max_{T \in U} \phi(T)$ ;
    Move  $T_b^*$  from  $U$  to  $L$ ;
  end for
until some stopping criterion.

```

ALGORITHM 2: Algorithm of pool-based feedback.

$v_{\max}$ : decide the granularity of searching space. We set it through values of training data.

$W$ : keep the particles motion inertia. We set the inertia weight at 0.8.

$c_1, c_2$ : represent accelerated weight of pushing each particle to  $pbest[]$  and  $gbest[]$ . We set both weights to the value of 2.

*Fitness Function.* We use fitness function to find two kinds of parameters in HMM models, that is, the optimized associated factor  $\delta_k$  and the  $S_k^{(c_i)}$ , which is the  $k$ th hidden states of the HMM model  $\lambda^{(c_i)}$ . Accordingly, the fitness function can be defined as follows:

$$\text{fitness}(\delta_k, S_k^{(c_1)}, \dots, S_k^{(c_u)}) = F_1\text{-Measure}, \quad (1 \leq k \leq u), \quad (10)$$

where  $F_1\text{-Measure}$  is the metrics (in Section 4.3) of classifiers' accuracy. There are a total number of  $7 \times u$  parameters that need to be searched. Since finding the whole set of parameters is time intensive by PSO, we divide the set of parameters into  $u$  independent parts. Seven parameters  $(\delta_k, S_k^{(c_1)}, \dots, S_k^{(c_u)})$ ,  $(1 \leq k \leq u)$ , are learned for each part according to the fitness function.

**3.3.2. Feedback.** Since it is time-consuming and expensive to manually label all the tweets, we introduce a feedback method that can automatically decide whether an unlabeled tweet shall be chosen and contained in a training data pool after it is assigned with an emotion by our HMM model. Unlike the strategy (Lewis and Catlett [21], Scheffer et al. [22]), which merely concerns the assigned emotions, a suitable strategy is to compute the entropy of a tweet  $T$  to identify the discrimination of emotions; that is, an emotion is more discriminating than all other emotions on the tweet  $T$ . Entropy is an information-theoretic measure and its formula is given as follows:

$$\phi(T) = -\sum_i P(c_i | T) \log P(c_i | T), \quad (11)$$

where  $P(c_i | T)$  means the probability of the tweet  $T$  recognized as emotion  $c_i$ . The less the  $\phi(T)$ , the more the

certainty about the tweet  $T$  on the emotion  $c_i$ . The algorithm of pool-based feedback is shown in Algorithm 2 to decide whether a tweet should be contained in the training dataset.

## 4. Experiments

**4.1. Dataset.** Since there are no public datasets of Chinese tweets associated with emotions, many studies employ emoticons to label tweets [7, 8, 10]. However, only a small percentage of Chinese users post tweets with emoticons. Besides, several emoticons cannot find their corresponding emotions. Hence, we build our own dataset as follows.

- (i) About 200,000 tweets were collected through Sina API.
- (ii) For each emotion, more than twelve seed terms were chosen for term-level feature extraction. We refer interested readers to Aman's annotation scheme [23] for seed term selection. For instance, a set of seed terms on happiness may contain "enjoy" and "pleased."
- (iii) Manually screening. Not all tweets are associated with a corresponding emotion. We asked ten students in computer science to choose good indicator tweet for the six emotions.

Notice that duplicate tweets are removed. Eventually, we selected 271, 159, 251, 229, 118, and 194 tweets on happiness, anger, surprise, fear, sadness, and disgust, respectively.

**4.2. Setup.** The procedure of emotion recognition for a tweet is shown as follows:

- (a) preprocess dataset;
- (b) build emotion lexicons; each lexicon contains a feature vector of terms extracted by feature extraction methods;
- (c) calculate feature vectors of the whole dataset;
- (d) optimize hidden states of each  $\lambda^{(c_i)}$  by PSO;
- (e) for each  $\lambda^{(c_i)}$  ( $c_i \in |C|$ ), calculate feature vector  $[y_1 \ y_2 \ \dots \ y_u]^{(c_i)}$  of  $T$  in emotion  $c_i$  and obtain

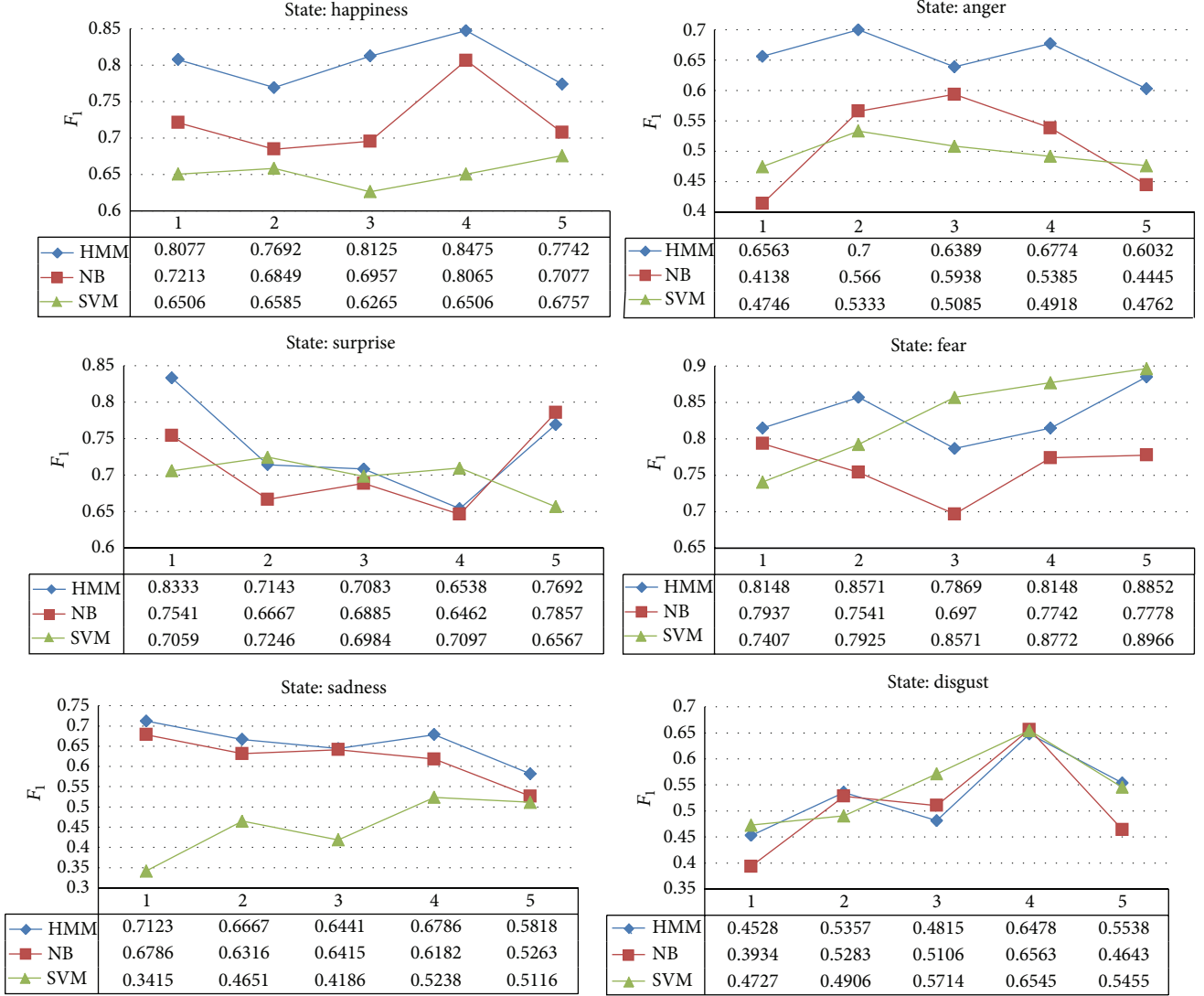


FIGURE 1: Performance comparison of three approaches on the six emotions, respectively. The x-axis means the number of test runs.

the output value  $P([y_1 \ y_2 \ \dots \ y_u]^{(c_i)} | \lambda^{(c_i)})$  through model  $\lambda^{(c_i)}$ ;

(f) return  $c^* = \operatorname{argmax}_{c_i \in C_i} \{P([y_1 \ y_2 \ \dots \ y_u] | \lambda^{(c_i)})\}$ .

To evaluate the performance of each of the HMM models, we randomly select several hand-annotated tweets. The test dataset contains 30 tweets for each emotion. Each test run is executed five times. We use the average results for our evaluation.

**4.3. Evaluation Metrics.** Precision, recall, and  $F$ -measure are the most commonly used evaluation methods for text classification tasks [7]. We employ the three metrics in our experiment. For each emotion  $c_i$ , precision and recall are defined as

$$\text{Precision} = \frac{a}{a+b}; \quad \text{Recall} = \frac{a}{a+c}, \quad (12)$$

where  $a$  represents the number of tweets that are correctly recognized;  $b$  represents the number of tweets that are falsely recognized as  $c_i$ ;  $c$  represents the number of tweets that are actually associated with  $c_i$ , but are recognized as another emotion.

To balance precision and recall rates,  $F_1$  is defined as

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (13)$$

#### 4.4. Evaluation

**4.4.1. Compare with Another Classifier.** We compared our approach with two well-known classifiers, that is, Naive Bayes and Support Vector Machine (SVM) [24]. They are often used for sentiment classification in literatures because of their easiness of implementation.

As shown in Figure 1, our HMM-based approach outperforms Naive Bayes and SVM on happiness, anger, and

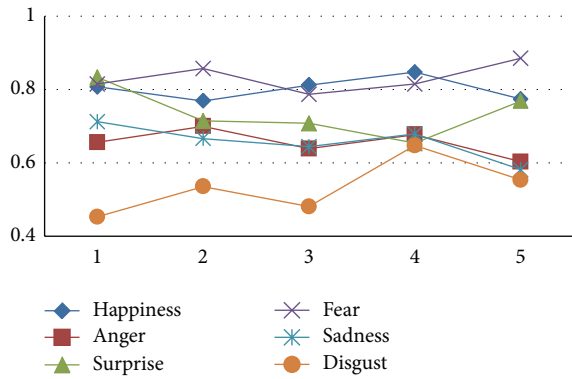


FIGURE 2: The accuracy of our HMM-based approach on the six emotions. The  $x$ -axis means the number of test runs.

sadness. The performances produced by all these classifiers have less difference on the other three emotions.

The results also show that, for the emotions of happiness, surprise, and fear, all the three classifiers get good performance. The  $F$ -measure is greater than 65%. Furthermore, the  $F$ -measure exceeds 78% on fear. None of the three classifiers recognize disgust accurately.

**4.4.2. The Comparison of Six Emotions.** Figure 2 shows the classification results of the six emotions using our HMM-based approach. We find that our approach gets the best accuracy over 76% on happiness and fear. It also gets a good performance on anger, surprise, and sadness with an average accuracy of 65%.

**4.4.3. Analysis of HMM Results.** We also attempt to analyze the reasons for false recognition on emotions. Twitters with specific characteristics may lead to false recognition using our HMM-base approach.

- (i) Some twitters may contain multiple emotions. For example, “Wow, I’m so smart!” (original sentence is in Chinese) contains both happiness and surprise. Multiple emotional tweets may cause false recognition. It also explains why our approach gets low  $F$ -score on disgust, which is often falsely recognized as anger.
- (ii) A number of tweets contain new words that can not be recognized.

In addition, puns and polysemous words are significant factors but rather difficult to be recognized. According to these reasons, our HMM-based approach can be improved concerning the above characteristics in our future experiments.

## 5. Conclusion

In this paper, we present an approach to extract features using the four methods, that is, MI, TF-IDF, CHI, and ECE. Our classifier is based on HMM, in which hidden states are found by PSO algorithm. Since manually labeling large-scale dataset is difficult, we employ the entropy to decide whether a new

unlabeled tweet shall be contained in the training dataset after being assigned an emotion using our HMM-based approach. The experimental results show that HMM outperforms SVM and NB, especially on happiness, anger, and sadness. In terms of the recognition precisions on the six emotions, HMM gets better performance on happiness and fear than on anger, surprise, and sadness.

In the future, we will optimize HMM to accurately recognize twitters associated with other emotions and automatically add new words for emotional seed terms selection. Moreover, self-adaptive mechanism will be implemented in our HMM model.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by Cuiying Grant of China Telecom, Gansu Branch (Grant no. lzudxcy-2013-3), Science and Technology Planning Project of Chengguan District, Lanzhou (Grant no. 2013-3-1), and Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (Grant no. 44th).

## References

- [1] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.
- [2] D. Quercia, J. Ellis, L. Capra, and J. Crowcroft, “Tracking ‘gross community happiness’ from tweets,” in *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW ’12)*, pp. 965–968, ACM, February 2012.
- [3] X. Lou, Y. Chai, H. Zan, R. Xu, Y. Han, and K. Zhang, “Research on micro-blog sentiment analysis,” in *Chinese Lexical Semantics*, pp. 466–479, Springer, Berlin, Germany, 2013.
- [4] A. Cui, H. Zhang, Y. Liu, M. Zhang, and S. Ma, “Lexicon-based sentiment analysis on topical chinese microblog messages,” in *Semantic Web and Web Science*, Springer Proceedings in Complexity, pp. 333–344, Springer, New York, NY, USA, 2013.
- [5] P. Ekman, “Universals and cultural differences in facial expressions of emotion,” in *Proceedings of the Nebraska Symposium on Motivation*, University of Nebraska Press, 1971.
- [6] J. Spencer and G. Uchyigit, “Sentimentor: sentiment analysis of twitter data,” in *Proceedings of the 1st International Workshop on Sentiment Discovery from Affective Data (SDAD ’12)*, M. Gaber, M. Cocea, S. Weibelzahl, E. Menasalvas, and C. Labbé, Eds., p. 56, CEUR, 2012.
- [7] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining,” in *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC ’10)*, Valletta, Malta, May 2010.
- [8] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” Project Report CS224N, Stanford University, Stanford, Calif, USA, 2009.

- [9] J. Zhao, L. Dong, J. Wu, and K. Xu, "MoodLens: an emoticon-based sentiment analysis system for Chinese tweets," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 1528–1531, August 2012.
- [10] Z. Yuan and M. Purver, "Predicting emotion labels for chinese microblog texts," in *Proceedings of the 1st International Workshop on Sentiment Discovery from Affective Data (SDAD '12)*, M. Gaber, M. Cocea, S. Weibelzahl, E. Menasalvas, and C. Labbé, Eds., pp. 40–47, CEUR, 2012.
- [11] R. A. Calvo and S. M. Kim, "Emotions in text: dimensional and categorical models," *Computational Intelligence*, vol. 29, no. 3, pp. 527–543, 2013.
- [12] S. D'Mello, A. Graesser, and R. W. Picard, "Toward an affect-sensitive autotutor," *IEEE Intelligent Systems*, vol. 22, no. 4, pp. 53–61, 2007.
- [13] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: the panas scales," *Journal of Personality and Social Psychology*, vol. 54, no. 6, pp. 1063–1070, 1988.
- [14] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, and Q. Liu, "HHMM-based Chinese lexical analyzer ICTCLAS," in *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing (SIGHAN '03)*, pp. 184–187, Sapporo, Japan, July 2003.
- [15] P. Frasconi, G. Soda, and A. Vullo, "Hidden Markov models for text categorization in multi-page documents," *Journal of Intelligent Information Systems*, vol. 18, no. 2-3, pp. 195–217, 2002.
- [16] P. Jaccard, *Nouvelles Recherches sur la Distribution Florale*, 1908.
- [17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.
- [18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [19] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, Springer, Berlin, Germany, 1998.
- [20] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
- [21] D. D. Lewis and J. Catlett, "Heterogenous uncertainty sampling for supervised learning," in *Proceedings of the International Conference on Machine Learning*, vol. 94, pp. 148–156, 1994.
- [22] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Advances in Intelligent Data Analysis*, vol. 2189 of *Lecture Notes in Computer Science*, pp. 309–318, Springer, Berlin, Germany, 2001.
- [23] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in *Text, Speech and Dialogue*, pp. 196–205, Springer, Amsterdam, The Netherlands, 2007.
- [24] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.





# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

