International Journal of Knowledge
and Language Processing
Volume 5, Number 2, **2014**

www.ijklp.org
KLP International ©2014 ISSN 2191-2734
pp.43—53

Study of Precise Sentiment Classification for Chinese Microblog based on Multilayered-Classifier

Tong Yixuan[1] Zhang Yangsen[2] Li Jingyu[3]

[1]Institute of Intelligence Information Processing
Beijing Information Science and Technology University
No. 12, East Qinghe-Xiaoying Road, Haidian District, Beijing, China
tongyixuanjames@163.com

[2]Department of Computer Science
Beijing Information Science and Technology University
zys@bistu.edu.cn

[3] Institute of Intelligence Information Processing
Beijing Information Science and Technology University
lijingyu_emily@126.com

Selected Paper from Chinese Lexical Semantic Workshop 2014

ABSTRACT. *Precise Chinese microblog sentiment classification became a hot topic recently. In this paper comparisons were made between different matches of Bayes Classifier and Sentiment-Word-Dictionary classifier in solving the problem of Precise Chinese microblog sentiment classification. We found that the combination of these two methods enhanced the performance of the whole system. Based on this study an advanced architecture for precise Chinese microblog sentiment classification system was put forward.*

Keywords: Bayes; Sentiment-Word-Dictionary; Multilayered-Classifier; Sentiment Classification

1. **Introduction.** Microblog is a medium of information sharing and transmitting. Users of microblog are able to build personal communities, link to the world every second. In August 2009, the Sina (a Internet company in China) microblog was formally published on the Internet, becoming the pioneer of Chinese microblog, since then Chinese people were getting to know the magic of microblog. In less than 3 years, the number of Tencent (anther Internet company) microblog users reached 507million; by the first half year in 2013, the number of Sina microblog users exceeded 536million. Weibo, the Chinese name for microblog, became really popular in China these years. We can be a part of microblog

communities with merely a cell phone. New ideas and technologies in this area bring us further remarkable experience, thus more people took part in this game every second. The users are mainly normal citizen. By analyzing the sentiment orientation of their words, options towards certain object can be discovered, so microblog sentiment classification is meaningful for commercial or political purpose.

While a lot of Work relating to English microblog sentiment classification has been carried out, Chinese microblog sentiment classification becomes a hot topic only recently. The classifier we are working on divides the microblog sentences into several classes. According to the number of categories, there are rough classification and precise classification. The former results in negative sentiment and positive sentiment. The latter, as we discussed in the paper, contains eight kind of sentiments. They are anger, disgust, fear, happiness, like, sadness, surprise and no sentiment. The rough classification already has achieved a correctness of more than 70%. On the other hand the precise classification, with a performance of hardly over 30%, is far from practical use. There are still lots of work to do.

The dataset applied in this paper was the one offered in NLPCC2013 (Natural Language Processing & Chinese Computing 2013) Chinese microblog sentiment classification task. We focused our effort on finding the right classify structure suitable for Chinese microblog sentiment classification purpose. We designed a classification system based on the combination of Native Bayes model and Sentiment-Word-Dictionary classifier, which we believe can outperform each single one.

2. **Related Work.** There are in general two approaches classifying the Chinese microblog sentiment. The first one is using supervised machine learning models. It is to choose features for a specific task, and train a model with tagged data. As widely believed, Native Bayes and SVM are the suitable classifiers in rough sentiment classification, which have achieved acceptable results; The second one is applying task related resource, for instance Sentiment-Word-Dictionary. It means to develop rules of using dictionaries manually, which can be useful under some circumstances. This chapter will discuss several related work all over the world.

2.1. **English Microblog Sentiment Classification.** Pang and Lee [1] studied common features for sentiment classification tasks. According to them, it was better to using the presence of words, in other words zeros and ones, instead of the TF-IDF values when applying bag of words feature set. Go et al [2,6] made a comparison among several features: unigrams, bigrams, unigrams and bigrams, POS(part of speech). They concluded that POS feature would make little contribution to sentiment classification purpose. Bigrams performed poorly due to data sparseness. The combination of unigrams and bigrams were very likely to get a better accuracy. They inherited the idea [3] of building tagged training set automatically with respect to the kind of emoticons a sentence contains. This saved lots of corpus annotation work, but on the other hand killed generality. Bermingham and

Smeaton [4] contrasted sentiment classification in microblog and in longer form documents (movie review, blog etc.). They argued that the shortness of the sentences would not significantly worsen the performance of the classifier. Surprisingly, punctuations were helpful classifying emotions. Pak and Paroubek [5] worked on feature selection. They tried entropy and salience (defined by themselves) to evaluate the contribution a feature can make to classification, and filtered out useless ones.

2.2. **Chinese Microblog Sentiment Classification.** Xu et al[7] constructed the affective lexicon ontology, they divided the emotions into seven types: anger, disgust, fear, happiness, like, sadness, surprise, and collected words with distinct emotions, then tagged the words both manually and automatically to ensure the correctness together with speed. Pang et al [8] built the training set in the similar way with Read [3], and tested unigrams and bigrams based classifier in Chinese microblog. They found these two features also effective under Chinese language environment. Yu [9] tried some brand new features: url, picture, negation. He made comparison between TF-IDF based SVM model and ordinary SVM model, and figured out that TF-IDF benefits little for Chinese microblog sentiment classification purpose. Lu [10] applied Sentiment-Word-Dictionary. He put forward methods to calculate sentiment orientation of a whole sentence based on the dictionary. Xie et al [11] studied multilayered classifier based on a various of features and achieved acceptable results.

3. **Methodology.** With normal approaches, the precise sentiment classification system would perform badly as expected. In 2013 NLPCC Chinese Microblog Sentiment Classification Task, the best system reached merely over 30% correctness. Under such circumstances, we studied multilayered classifier to see if it is better compared with ordinary methods. Experiments were made on combinations of Native Bayes model and Sentiment-Word-Dictionary model. These were the two elements of our classifier.

3.1. **Native Bayes.** The Native Bayes model is based on Bayesian theory. The class of a microblog which is assigned to δ can be calculated as follows:

$$\delta = \arg\max_{S_j} [P(S_j) \prod_{c_i \in C} P(c_i \mid S_j)] \tag{1}$$

Where c_i is a feature extracted from a microblog post. Take unigrams as example, c_i will be a single word from the sentence. $P(S_j)$ is the possibility of a microblog sentence being sentiment S_j. $P(c_i \mid S_j)$ is like hood of c_i being present when the sentiment of the blog is S_j. $P(S_j)$ and $P(c_i \mid S_j)$ are learnt from train process. We use add-1 smoothing. Native Bayes is a very simple model, but it works surprisingly well in solving sentiment classification problems. The training and testing of this model can be really fast compared with other models such as SVM.

The features we used in this paper are unigrams and bigrams, which are suitable for sentiment classification tasks [2, 6].

3.2. **Sentiment-Word-Dictionary Classifier.** The Sentiment-Word-Dictionary we use in our experiment is the affective lexicon ontology (Xu et al.,2008). It is a precisely tagged in two levels. There are 27466 sentiment words in this dictionary, which are divided into 7 categories, and further into 21 small classes. Another parameter emotional intensity indicates the strength of the emotion. We use emotion category and emotion intensity in our model. Let d_i be the emotion intensity vector $\begin{pmatrix} a_1 \\ \vdots \\ a_7 \end{pmatrix}$ for an emotion word. The maximum of parameter i is number of emotion words in a microblog post. $a_1, a_2 \ldots a_7$ are the emotion intensity of each category. In the affective lexicon ontology, every word is assigned to a unique category, which means among $a_1, a_2 \ldots a_7$, only one is nonzero value. We add up all the vectors in a microblog post as follows:

$$\mathrm{D}=\sum d_i \tag{2}$$

The max dimension in D indicates the most intense emotion, which we believe is very likely to be the right emotion category.

3.3. **Multilayered-Classifier.** The Multilayered-Classifier we discussed in this paper consist two level of classifiers: the rough classifier and the precise classifier. We made experiments on three different combination. As shown in Table 1.

TABLE 1. THE COMBINATIONS OF CLASSIFIERS

Combination Name	Rough Classifier	Precise Classifier
MultiNbm	Native Bayes	Native Bayes
NBM-SWDC	Native Bayes	SWDC
SWDC-NBM	SWDC	Native Bayes

The abbreviation SWDC refers to Sentiment-Word-Dictionary Classifier. To make comparisons, we also run tests on simple SWDC and Native Bayes.

The training process can be described as Algorithm 1.

ALGORITHM 1. TRAINING PROCESS OF THE MULTILAYERED CLASSIFIER

Step-1: Preprocess, segment the microblog post into words and filter out useless or harmful words and punctuations. Extract features if necessary.
Step-2: If the tag of the microblog post is 'none', turn to step 5.
Step-3: Train the rough classifier with tag 'Y'
Step-4: Train the precise classifier with the original tag. Turn to step 6.
Step-5: Train the rough classifier with tag 'N'
Step-6: End

The Sentiment-Word-Dictionary Classifier need no training. Thus, do nothing when it's turn to train a Sentiment-Word-Dictionary Classifier. We design universal programming interfaces for all classifier to reduce repetitive coding. Literally, the training interface for SWDC does nothing.

The testing process can be described as Algorithm 2.

ALGORITHM 2. TESTING PROCESS OF THE MULTILAYERED CLASSIFIER

```
Step-1: Preprocess, segment the microblog post into words and filter out useless or harmful words and punctuations. Extract features if necessary.
Step-2: Run classification with the rough classifier. If the result turns out to be 'N', go to step 4.
Step-3: Run classification with the precise classifier. Take the output as the final result. Go to step 5.
Step-4: Take 'none' as the final result.
Step-5: End
```

3.4. **Clause and Sentence Level Training.** In 2013 NLPCC Chinese Microblog Sentiment Classification Task, the data set given is tagged by both clause-level and sentence-level, so it's possible to compare the performance of models trained by clause-level tags and the ones trained by sentence-level tags. Clause-level tagged data give more detailed information, and the scale of it is larger which is likely to improve machine learning models. The clause-level trained classifier results in emotion classification of clauses, so to make a comparison we have to find a way to determine sentence emotion category based on emotions of clauses. We built a rule-based classifier according to statistics shown in table 2.

TABLE 2. RELATION BETWEEN SENTENCE EMOTION AND CLAUSE EMOTION

Percentage of the same	First Clause Emotion	Last Clause Emotion	Majority Clause Emotion
Sentence Emotion	28.5%	85.0%	93.9%

We can see the emotion of the sentence is very likely to be the emotion majority of clauses shared. Based on this discovery, we present Algorithm 3 to solve sentence emotion with clause emotions.

ALGORITHM 3. CALCULATING SENTENCE EMOTION

```
Step-1: Count the emotion of each clause. Sort the result.
Step-2: Get the largest two numbers from the sorted list. If these two numbers were equal, Turn to step 4.
Step-3: Take the emotion corresponding to the largest number. Turn to step 7.
Step-4: If the emotion of the last clause is 'none', turn to step 6.
Step-5: Take the emotion of the last clause as result. Turn to step 7.
Step-6: Take the emotion of the second last clause as result.
Step-7: End
```

We tested the rules using the training set (4000 sentences) offered by NLPCC2013 Chinese Microblog Sentiment Classification Task. It achieved 94.7% correctness. In the later experiments we adopted this strategy.

4. **Results and Discussion.** In this section, we made comparisons among different combination of classifiers, and between Sentence-Level trained systems and Clause-Level trained ones.

4.1. **Dataset and Preprocess.** The data set we used in the paper is from NLPCC2013 Chinese Microblog Sentiment Classification Task. It contains 10000 microblog posts. Each sentence and clause are tagged with one of eight labels: anger, disgust, fear, happiness, like, sadness, surprise, none (no emotion). The distribution of each emotion is as shown in Table 3.

TABLE 3. SENTIMENT DISTRIBUTION

	None	Disgust	Happiness	Like	Fear	Anger	Sadness	Surprise
Percentage	49.25%	9.69%	11.06%	15.25%	0.9%	4.05%	7.59%	2.21%

We segmented the sentences by ICTCLAS2013, then filtered out the stop words. The stop words collection is like Table 4.

TABLE 4. STOP WORDS

Chinese Words:	转发此微博(repost a microblog), 回复(reply a microblog)
punctuations:	@ . * [: ,] '
Urls:	all urls

4.2. **Evaluation Standards.** We evaluated the classifier with precision, recall and F value.

$$\text{Precision} = \frac{Correct(x)}{\mathrm{Re}\,call(x)} \tag{3}$$

$$\text{Recall} = \frac{Correct(x)}{Label(x)} \tag{4}$$

$$\text{F} = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Re}\,call}{\mathrm{Pr}\,ecision + \mathrm{Re}\,call} \tag{5}$$

$Correct(x)$ is the correct classifications for emotion x by the classifier. $\mathrm{Re}\,call(x)$ is the number of sentences assigned to emotion X by the classifier. $Label(x)$ is the number of sentence being tagged as emotion x. Usually, precision and recall won't rise simultaneously. They are two aspects of the performance. Pursuing only precision or recall means little, thus we invite F value to take both these two factors into consideration. Parameter x is the emotion category, which excludes the class "none"(indicating no emotion). The distribution of the data set is not balanced. Nearly 50% the sentences are tagged as none, so to build a classifier assigning all sentences to none class will achieve almost 50% correctness, which is senseless. By excluding the evaluation of class none we will get a meaningful result.

We used 10-fold cross-validation in the training and testing process. That is to separate the tagged data set into 10 groups. One by one, take one group as the testing set, and the rest nine groups as the training set. We calculated the average value and the standard deviation (as formula 6) of the ten results.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (6)$$

The Native Bayes model, the SWDC classifier and the evaluation functions applied in this experiment were implemented by ourselves using C#. We adopted add-1 smoothing for Native Bayes model, and the smooth factor was 0.01.

4.3. **Results Analysis.** The sentence-level trained results are in table 5.

TABLE 5. SENTENCE-LEVEL TRAINED RESULTS

Combination Name	Precision	Recall	F-Value
Nbm	0.362	0.022	0.042
SWDC	0.207	0.265	0.233
SWDC-Nbm	0.209	0.291	0.244
MultiNbm	0.309	0.181	0.228
Nbm-SWDC	0.307	0.203	0.245

In the table, Nbm is the simple Native Bayes, and SWDC is the Sentiment-Word-Dictionary classifier. Nbm achieves the best precision, but the recall of it is unbearably low. On contrast, the Recall of SWDC is far better than Nbm. Finding a way to merge advantages from both sides was the reason we come up with the idea of combining these two in the first place. We can see that SWDC-Nbm and Nbm-SWDC outperformed single-layered classifiers.

Surprisingly, the recall of SWDC-Nbm is better than SWDC (SWDC was supposed to achieve the best recall). To explain the reason, we need to introduce MultiSWDC, which means both the rough classifier and precise classifier are SWDC. It works completely same with SWDC, so we look on them as equal. MultiSWDC and SWDC-Nbm will recruit the same set in rough classification stage. The reason MultiSWDC (the same as SWDC) gets a lower recall than SWDC-Nbm, is that Nbm(trained merely by small part of the training set) works better than SWDC in precise classification stage. Furthermore, we think that the accuracy of Sentiment-Word-Dictionary is limited by the fact that each word is assigned to a single emotion. If we express the information learned by a Native Bayes models as vectors, each word has its own vector with every dimensions could be nonzero value. We manage to extract the vector of the word "果断"(pronunciation: guo duan, meaning: decisive) from the Native Bayes model, and the one from Sentiment-Word-Dictionary. They were listed in Table 6.

TABLE 6. INFORMATION VECTOR OF "果断"

	None	Disgust	Happiness	Like	Fear	Anger	Sadness	Surprise
Native Bayes	6	4	1	3	0	2	1	0
SWDC	0	0	0	5	0	0	0	0

The Native Bayes vector records the presence count of "果断" in each emotion category("果断" is a rare word in microblog posts). The SWDC vector reflects the emotion intensity and emotion classification according to the Sentiment-Word-Dictionary. When performing a rough emotion classification of a sentence containing "果断", these two vector were likely to lead to the same result, but precise classification on the other hand would end differently. The Native Bayes vector contains more information about the possibility of this word being present in sentences with various emotions, thus it is more comprehensive.

The clause-level trained results are in table 7.

TABLE 7. CLAUSE-LEVEL TRAINED RESULTS

Combination Name	Precision	Recall	F-Value
Nbm	0.444	0.071	0.123
SWDC	0.167	0.219	0.189
SWDC-Nbm	0.183	0.378	0.247
MultiNbm	0.345	0.212	0.262
Nbm-SWDC	0.310	0.105	0.157

The number of the clauses is more than 30000, thus Clause-Level tagged data set contains more information. Machine Learning methods are supposed to improve by enlarging the training data set. As expected, MultiNbm achieves the highest F-value. It is very likely that given an even larger training set, MultiNbm will work better. The performance of SWDC gets worse suggesting that SWDC is not suitable for precise clause emotion classification.

The Standard deviation we invited in this experiment is a measurement of stability. Along the 10-fold cross-validation process, the training set and the testing set changes, a smaller standard deviation implies the classifier suffers less due to these changes, or in other words it is more data-independent. Usually, classifiers we trained in NLP field are data-dependent, which means a classification system may work well on a dataset but pretty bad on another. To evaluate the data-dependency, we presents SF (the stability factor), which is shown in formula 7. The reason we divides the standard deviation of precision with average precision is to avoid a small precision leading to a small standard deviation of precision. The results are listed in the following tables.

$$SF=\frac{Standard\_deviation\_of\_Precision}{Average\_Precision}+\frac{Standard\_deviation\_of\_Recall}{Average\_Recall} \quad (7)$$

TABLE 8. STANDARD DEVIATIONS OF SENTENCE-LEVEL TRAINED CLASSIFIERS

Combination Name	Standard deviation of Precision	Standard deviation of Recall	SF
Nbm	0.160	0.007	0.78637
SWDC	0.034	0.030	0.28040
SWDC-Nbm	0.039	0.036	0.31136
MultiNbm	0.082	0.014	0.34551
Nbm-SWDC	0.060	0.022	0.30781

TABLE 9. STANDARD DEVIATIONS OF CLAUSE-LEVEL TRAINED CLASSIFIERS

Combination Name	Standard deviation of Precision	Standard deviation of Recall	SF
Nbm	0.063	0.009	0.27130
SWDC	0.017	0.015	0.17348
SWDC-Nbm	0.009	0.017	0.09951
MultiNbm	0.046	0.008	0.17558
Nbm-SWDC	0.046	0.015	0.30166

The feature distribution in sentence-level tagged data set is different compared with that in clause-level tagged data set. Due to the change of the training set, Nbm is the most unstable classifier in sentence-level stability experiment. SWDC on the other hand never learns new information, thus it tended to be more stable. By constructing Multi-Layered classifiers, even better stability can be achieved. SWDC-Nbm gets the best score in clause-level stability experiment, which implies that by combining two classifiers, we can get a more accurate and stable classification system. The SF can be looked on as the second important factor (the most important one is the F-value) to evaluate a classification system. Furthermore, these experiments infers that Nbm will be more steady if it is fully trained, suggesting a way of measuring the training completeness for a machine learning model.

Based on F-value and stability factor, we can elect the classifiers for practical usage. We will discuss F-value first. In Sentence-Level accuracy experiment, SWDC-Nbm and Nbm-SWDC are the most accurate two classifiers. However, in Clause-Level accuracy experiment, Nbm-SWDC works pretty bad, MultiNbm together with SWDC-Nbm become one of the best two. We concludes that Nbm-SWDC is heavily data-dependent. SWDC-Nbm and MultiNbm are advisable. SWDC-Nbm works well for a small training set. MultiNbm will perform better than SWDC-Nbm if the training set is large enough. Considering stability factor, the SWDC-Nbm is the most stable classifier of all, and the MultiNbm is likely to be stable if trained enough. Single-layered Native Bayes is not sufficiently trained in all experiments.

5. **Conclusions.** In this paper, we studied multi-layered classifiers. Based on different combinations of Native Bayes model and Sentiment-Word-Dictionary classifier, we made comparison experiments to measure the accuracy and stability of these classifiers.

Multi-layered system could achieve results with more favorable F-value and stability factor (SF) according to experiment results. Especially, the MultiNbm and SWDC-Nbm are the best two. There are two rules of selecting the suitable one between these two. The first rule is if the training data set is large enough (much larger than 10000 objects), using MultiNbm will be a good idea, otherwise it is better to apply SWDC-Nbm as the classifier. The other rule is if the cost of classification error for one category (for example the 'none' class in our experiment) is low, MultiNbm will suit the purpose better.

We found in our experiments that the Sentiment-Word-Dictionary classifier might suffer from inherent shortcoming when dealing with Precise Chinese microblog emotion classification tasks. Dictionaries assigned a sentiment word to a single category neglected many possibilities. It was not as accurate as machine learning based classifiers, but as we experimented, Sentiment-Word-Dictionary classifier achieved good recall rate. It could be meaningful under some circumstances.

Further studies needs to be done in the future. First, in the training process, we didn't managed to make full use of the data set. The precise classifier was trained by sentences tagged with seven categories, besides 'none'. We will find ways to avoid this problem. Second, the F-values of combinations we tried were still low. We needs to find better features for precise emotion classification task to retrieve overlooked information such as word sequence, negations. What's more, we like to try some new thoughts, for example to extract deep-seated features using artificial neural network.

Acknowledgment. This work is funded by the Natural Science Foundation of China(NSFC, Grant No.61070119, No.61370139), the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality(Grant IDHT20130519) and the Beijing Municipal Education Commission Special Fund(Grant PXM2013_014224_000042，PXM2014_014224_000067). We are grateful for these economic support. Special thanks to my supervisor, Yangsen Zhang, for his advice and guidance.

REFERENCES

[1] B. Pang and L. Lee, Opinion mining and sentiment analysis, Foundation and Trends in Information Retrieval, 2(1-2):1–135, 2008.

[2] Alec Go, Richa Bhayani, and Lei Huang, Twitter Sentiment Classification using Distant Supervision, Processing (2009).

[3] Jonathon Read, Using emoticons to reduce dependency in machine learning techniques for sentiment classification, In Proceedings of ACL-05, 43nd Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pages 43-48, 2005.

[4] Adam Bermingham and Alan Smeaton, Classifying Sentiment in Microblogs: Is Brevity an Advantage,

Proceeding CIKM '10 Proceedings of the 19th ACM international conference on Information and knowledge management, pages 1833-1836, 2010.

[5] Alexander Pak and Patrick Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining, Proceedings of LREC, 2010.

[6] Bo Pang and Lillian Lee, Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP, pages. 79--86, 2002.

[7] Xu Linhong, Lin Hongfei, Pan Yu, RenHui and Chen Jianmai, Constructing the Affective Lexicon Ontology, Journal of the China Society for Scientific and Technical Information, 27(2): 180-185, 2008. [In Chinese]

[8] Pang Lei, Li Shoushan and Zhou Guodong, Sentiment Classification Method of Chinese Micro-blog Based on Emotional Knowledge, Computer Engineering, 2012. [In Chinese]

[9] Yu Jianping, Micro-Blog Sentiment Analysis Based on Semantic Sentiment Space Model [D], Guangzhou China: Jinan University, 2012. [In Chinese]

[10] Lu Yujie, Chinese Twitter Sentiment Analysis [D], Shanghai China: East China Normal University, 2013. [In Chinese]

[11] Xie Lixing, Zhou Ming, and Sun Maosong, Hierarchical Structure Based Hybrid Approach to Sentiment Analysis of Chinese Micro Blog and Its Feature Extraction, Journal of Chinese Information Processing, 2012. [In Chinese]