J Inf Process Syst, Vol.14, No.3, pp.727~739, June 2018
https://doi.org/10.3745/JIPS.04.0076

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Microblog Sentiment Analysis Method Based on Spectral Clustering

Shi Dong*,**, Xingang Zhang***, and Ya Li*

Abstract

This study evaluates the viewpoints of user focus incidents using microblog sentiment analysis, which has been actively researched in academia. Most existing works have adopted traditional supervised machine learning methods to analyze emotions in microblogs; however, these approaches may not be suitable in Chinese due to linguistic differences. This paper proposes a new microblog sentiment analysis method that mines associated microblog emotions based on a popular microblog through user-building combined with spectral clustering to analyze microblog content. Experimental results for a public microblog benchmark corpus show that the proposed method can improve identification accuracy and save manually labeled time compared to existing methods.

Keywords

Machine Learning, RDM, Sentiment Analysis, Spectral Cluster

1. Introduction

With the development of online social networks, people have begun to express their feelings, emotions, and attitudes online. Microblogs, such as Twitter and Sina Weibo, constitute a popular type of social networking platform. Thus, when an exciting event occurs, copious amounts of news and public opinions increase user data. The notion of how to mine useful data may provide implications for government officials and companies. Many methods have been proposed to mine big data from social media (e.g., microblogs), of which sentiment analysis (SA) is a popular approach.

SA is also referred to as view mining, a means of mining people's feelings and attitudes from texts. SA is divided into three levels: document-level [1], sentence-level [2], and aspect-level [3]. Medhat et al. [4] noted that document-level SA aims to classify an opinion document as expressing either a positive or negative opinion or sentiment. The level considers the whole document a basic information unit (i.e., addressing one topic). Sentence-level SA aims to classify the sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective; if it is subjective, then sentence-level SA will determine whether the sentence expresses a positive or negative opinion. Aspect-level SA seeks to classify sentiment with respect to the specific aspects of entities. The first step is to identify the

Manuscript received April 26, 2016; first revision November 11, 2016; accepted February 1, 2017.
Corresponding Author: Shi Dong (njbsok@gmail.com)
* School of Computer Science and Technology, Zhoukou Normal University, Zhoukou, China (njbsok@gmail.com, 37227144@qq.com)
** School of Computer Science and Technology, HuaZhong Universtiy of Science and Technology, Wuhan, China (njbsok@gmail.com)
*** School of Computer and Information Technology, Nanyang Normal University, Nanyang, China (zxg550@163.com)

entities and their aspects. Opinion holders can offer different opinions regarding various aspects of the same entity. Because social media texts possess common characteristics, such as limited length and informal expression, SA can be challenging. Existing SA methods tend to use supervised machine learning (ML) to train the data, after which an identification model is constructed to identify sentiment data; however, these often rely on manual labels marked in advance. To improve the performance of microblog sentiment identification tasks, the present authors adopt a SA method based on semi-supervised spectral clustering to analyze and identify microblog emotion text. The proposed method only requires a few artificial labels, thus reducing the workload and improving the efficiency of SA.

The rest of this paper is organized as follows. Section 2 discusses related work on SA. Section 3 proposes spectral clustering SA models. Section 4 introduces the evaluation metric. Experimental results are provided in Section 5. Finally, conclusions and future work are presented in Section 6.

2. Related Work

Because ML methods are unlikely to be affected by dictionary size and updates, an increasing number of researchers focusing on SA have turned to such methods to classify sentiment in texts. Pang et al. [5] introduced ML methods in sentiment classification and adopted a naive Bayes classification, maximum entropy classification, and support vector machines (SVMs) to classify a movie-review corpus. Paltoglou and Thelwall [6] studied document representations with SA using term weighting functions adopted from information retrieval and adapted to classification. The proposed weighting schemes were tested with several publicly available datasets, many of which repeatedly demonstrated significant increases in accuracy using these schemes compared to other state-of-the-art approaches. Jin et al. [7] proposed a novel and robust ML system for opinion mining and extraction. Xu et al. [8] used a naive Bayes and maximum entropy model to mine Chinese web news and complete automatic classification of emotion-related content, combining a special microblog dictionary with a traditional emotional dictionary. However, ML applications are limited given the need for a correctly labeled corpus as a basis for training and learning. When the difference between the object sample and the training sample is large, results are inadequate. Yin, Pei, et al. [9] mainly focused on improving sentiment classification in Chinese online reviews by analyzing and improving each step in supervised ML. The experimental results indicated that part of speech, number of features, evaluation domain, feature extraction algorithm, and SVM kernel function exerted great influences on sentiment classification, whereas the number of training corpora had little impact. Da Silva et al. [10] proposed an integrated classifier to analyze Twitter emotion by integrated naive Bayes, SVM, random forest, and logistic regression approaches; experimental results showed that the ensemble classifier could improve emotional classification accuracy. Zhang et al. [11] suggested a new SA method that combined text and image information using the similarity neighbor classification model to classify emotions and effectively improve classification accuracy. Jiang et al. [12] put forth an improved SA model constructed to manage more unlabeled data, establish emotional space, and effectively grab emotional keywords in SA to improve efficiency. Barborsa and Feng [13] presented an effective and robust sentiment detection approach for Twitter messages, using biased and noisy labels as input to build the models. Pang et al. [14] used emotional words and emoticons to filter non-marked microblogging corpus, constructed the corpus, and then trained the resultant automatic annotation corpus as a training set to build a classifier

for microblog emotion-related text and classify emotional polarity in microblogging text. Liu et al. [15] constructed dictionaries of sentiment words, internet slang, and emoticons, respectively, and then implemented SA algorithms based on phrase paths and multiple characteristics of emotional tendency in microblog topics. Using microblog forwarding, comments, sharing, and similar behaviors, this algorithm could be optimized in the future based on multiple characteristics. Go et al. [16] used Twitter to collect training data and perform a sentiment search to construct corpora by using emoticons to obtain positive and negative samples followed by the application of various classifiers; however, this method demonstrated poor performance across three classes (i.e., negative, positive, and neutral). Liu et al. [17] presented a novel model called the emoticon-smoothed language model to address this issue. The basic idea is to train a language model based on manually labeled data and then use noisy emoticon data for smoothing. Che et al. [18] applied a discriminative conditional random field model with special features to compress sentiment sentences automatically; experimental results highlighted the effectiveness of the feature sets used for sentiment sentence compression (Sent_Comp) and the effectiveness of the Sent_Comp model applied in aspect-based sentiment analysis. Jiang et al. [2] incorporated target-dependent features and took related tweets into consideration. Dong et al. [19] proposed a set-similarity joint-based semi-supervised approach, which joined nodes in unconnected sub-graphs by cutting the flow graph with the Ford-Fulkerson algorithm into positive and negative sets to correct incorrect polarities predicted by min-cut-based semi-supervised methods. Although text-based SA has achieved notable success to this point, SA in Chinese texts still suffers from unresolved problems.

3. Spectral Clustering SA Model

Traditional research on SA methods generally focus on supervised ML methods. This section introduces a new semi-supervised SA method based on spectral clustering to construct a clustering model. Spectral clustering is a clustering method based on graph theory, which makes use of the spectrum (i.e., eigenvalues) of the similarity matrix of data to perform dimensionality reduction before clustering on fewer dimensions. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset.

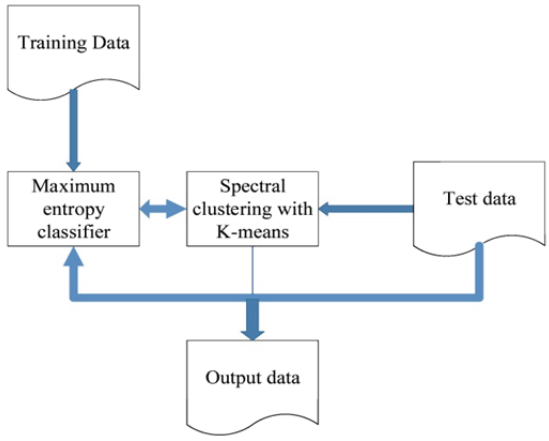


Fig. 1. Spectral clustering sentiment analysis model.

Fig. 1 illustrates that the model is a traditional semi-supervised ML approach that differs from other semi-supervised methods and adopts spectral clustering to improve the cluster results. The classifier adopts sentiment sentence classifier-based maximum entropy [12]. The analysis procedure is as follows. First, training data are used as classifier input to train the maximum entropy classifier. Next, spectral clustering with k-means is employed to improve the classifier and incorporate test data into the classifier. Finally, output data is obtained as a labeled sentence. This section mainly focuses on the spectral cluster method and related information.

3.1 Graph Partition

A *graph partition* is defined as data represented in the form of a graph $G=(V,E)$, with V vertices and E edges, such that it is possible to partition G into smaller components with specific properties. For instance, a k-way partition divides the vertex set into k smaller components. A good partition is defined as one in which the number of edges running between separate components is small. Therefore, the size of a sub-graph is nearly sufficient, and the weight of the cutting edge reaches the minimum. A graph partition can be considered a constrained optimization problem involving how to divide each point into a sub-graph. Unfortunately, when choosing a variety of objective functions, the optimization problem is often NP-hard. A relaxation method can be helpful in solving this problem, especially in transforming a combinatorial optimization problem into a numerical optimization problem that can then be solved in polynomial time before finally being restored by a threshold when restoring the division. A similar method to k-means may also apply with instructions. The related definition is as follows:

Given an undirected distance graph $G(V,E)$, let G be a graph with V vertices and E edges. If an edge belongs to a sub-picture, then two vertices of the edge are included in the sub-graph. Assume two different endpoints exist from edge E, where the weight of E is denoted as $w_{i,j}$. For an undirected graph, $w_{i,j}= w_{j,i}$ and $w_{i,j}= 0$. The graph partition method is referred to as *cut*, defined as all end points not existing on the same side of the sub-sum of the weights and the figure (i.e., a loss of function of the partition plan, which is intended to be as small as possible). This paper takes an undirected graph as an example; assume the original undirected graph G, which is divided into G_1 and G_2, is denoted as

$$cut(G_1, G_2) = \sum_{i \in G_1, j \in G_2} w_{i,j} \tag{1}$$

Laplacian matrix

Assume undirected graph G is divided into two sub-graphs, G_1 and G_2. The vertices of G are $n = |V|$, and q is an n-dimensional vector, which can then be denoted as

$$\begin{aligned} cut(G_1, G_2) &= \sum_{i \in G_1, j \in G_2} w_{i,j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}(q_i - q_j)^2}{2(c_1 - c_2)^2} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} -2w_{i,j}q_i q_j + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}(q_i^2 + q_j^2) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} -2w_{i,j}q_i q_j + \sum_{i=1}^{n} 2q_i^2 (\sum_{j=1}^{n} w_{i,j}) \\ &= 2q^T (D - W)q \end{aligned} \tag{2}$$

where D is the diagonal matrix. Diagonal elements are defined as follows:

$$D_{i,i} = \sum_{j=1}^{n} w_{i,j} \tag{3}$$

where W is a weight matrix, because $w_{i,j} = w_{j,i}$,and $w_{i,j} = 0$. L is a Laplacian matrix, defined as $L=D-W$. From there, we obtain the following:

$$q^T Lq = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}(q_i - q_j)^2 \tag{4}$$

If the gravity ownership is non-negative, then $q^TLq \geq 0$, indicating the Laplacian matrix is a semi-positive definite matrix. When no connectivity exists, the eigenvalues of L are 0 and the corresponding feature vectors are $[1, 1, ..., 1]^T$. Hence, if an undirected graph G is partitioned into two sub-diagrams, then one is itself, the other is empty, and the cut is 0. Thus, we obtain the following Eq. (5):

$$cut(G1, G2) = \frac{q^T Lq}{(c_1 - c_2)^2} \tag{5}$$

Eq. (5) indicates that the minimization cut partition problem is converted to a minimization quadratic function q^TLq; that is, it is a problem from seeking the relaxation of discrete values into continuous real values. The Laplacian matrix can better represent a graph; any such matrix corresponds to an undirected graph of non-negative weights, and the Laplacian matrix should meet the following conditions:

1. L is a symmetric, positive, semi-definite matrix to ensure that all eigenvalues are greater than or equal to 0;
2. The matrix L has a unique characteristic value of 0 and corresponding feature vectors of [1,1,...,1], reflecting a graph partition: A sub-graph contains all the endpoints of the graph, and another sub-graph is empty.

3.2 Partition Method

Several methods are available for graph partitioning, including minimum cut, ratio cut, and normalized cut; this paper uses the normalized cut method, which measures the sub-graph according to each degree sum of the endpoint from the sub-graph. Let d_1 be the degree sum of G_1, defined as

$$d_1 = \sum_{i \in G_1} d_i \tag{6}$$

Let d_2 be the degree sum of G_2, defined as

$$d_2 = \sum_{i \in G_2} d_i \tag{7}$$

Then, the objective function is

$$obj = cut(G_1, G_2) * (\frac{1}{d_1} + \frac{1}{d_2}) = \sum_{i \in G_1, j \in G_2} w_{i,j} * (q_i - q_j)^2 \tag{8}$$

However, relaxation of the original problem is based on the following:

$$\begin{aligned} &min \quad q^T Lq \\ &subject \quad to \quad q^T De = 0 \\ &\qquad\qquad\quad q^T Dq = 1 \end{aligned} \tag{9}$$

The generalized Rayleigh quotient is expressed as

$$R(L,q) = \frac{q^T Lq}{q^T Dq} \tag{10}$$

The problem can be converted to obtain the eigenvalues and eigenvectors in the features system:

$$\begin{aligned} &Lq = \lambda Dq \\ &\Leftrightarrow Lq = \lambda D^{\frac{1}{2}} D^{\frac{1}{2}} q \\ &\Leftrightarrow D^{-\frac{1}{2}} LD^{-\frac{1}{2}} D^{\frac{1}{2}} q = \lambda D^{\frac{1}{2}} q \\ &\Leftrightarrow L'q' = \lambda q' \end{aligned} \tag{11}$$

$$L' = D^{-\frac{1}{2}} LD^{-\frac{1}{2}}, q' = D^{\frac{1}{2}} \tag{12}$$

In Eq. (12), $Lq = \lambda Dq$ has the same eigenvalues as $L'q' = \lambda q'$, and a relationship exists between the feature vectors corresponding to the eigenvalues $q' = D^{½}q$. Therefore, after the eigenvalues and eigenvectors are obtained in Eq. (12), each feature vector $D^{-(1/2)}$ can be multiplied. Then, the eigenvectors for $Lq = \lambda Dq$ are got.$L' = D^{-(1/2)} LD^{-(1/2)}$ constitute a normalized Laplacian matrix.

3.3 Spectral Cluster Method (SASC) Algorithm

This paper proposes an SASC. To mine sentiment sentences, nodes are considered sentences. Sentiment sentence features are sentiment patterns. A maximum entropy [12]-based sentiment sentence classifier is used to predict primary polarities. Then, a flow graph of sentences can be constructed using these candidate sentences. The proposed method (a normalized spectral clustering algorithm) is described as follows:

Algorithm Input: a sample matrix S and a similar number of classes to be clustered k.
Algorithm Output: O.

Step 1: Establish the right weight matrix W based on the matrix S and triangular matrix D;
Step 2: Establish Laplacian matrix L;
Step 3: Use maximum entropy to predict the primary polarities, ME(L);
Step 4: Compute k eigenvalues and the corresponding eigenvectors of Matrix L, the minimum of the eigen values must be 0, and the corresponding feature vector is $[1, 1, ..., 1]^T$;
Step 5: Considering k feature vectors as a new matrix, the number of rows is the number of samples, and the number of columns is k; Dimensionality reduction is done from N to k;
Step 6: Use k-means clustering algorithm to obtain the k cluster.

Step 7: Compare unlabeled sentence with labeled sentence by maximum entropy; if equal, add the unlabeled sentence into the labeled sentence that is the same as ME(L).

Step 8: Output labeled sentence.

Algorithm 1: SASC algorithm

```
Input: S, k;
Output: O;
1 Begin
2  for 1 to k do
3   construct weight matrix W;
4     construct triangular matrix D;
5     establish Laplacian matrix L;
6     O'=ME(L);
7  obtain eigenvectors and eigenvalues;
8      k=k+1;
9      while k!=0 do
10    if L' eigenvalues!=null then
11    Lnew=Lk;
12    out=kmeans(k);
13   if O'=out then
14    O=O';
15   else
16    return error;
17  return O;
18 End
```

As shown in Algorithm 1, the SASC algorithm uses maximum entropy to construct the classifier and label the sample data. Nigam et al. [20] pointed out that the maximum entropy method performs better than naive Bayes. In addition, the SASC algorithm introduces spectral clustering to cluster microblog sentiment data. Thus, a small amount of labeled data and large amount of unlabeled data apply. When using k-means to compare the cluster with the labeled data, the cluster data will be labeled if requirements are met. The SASC algorithm can reduce sample data to train the classifier and improve identification efficiency.

4. Performance Analysis

4.1 Evaluation for Sentiment Sentence Extraction

This paper employs the routine evaluation standard to verify the effectiveness of the proposed algorithm. The following three evaluation criteria apply:

TP (true positive): The sentiment sentence *S* is correctly classified as *S*, which is a correct classification result;

FP (false positive): The sentiment sentence outside *S* is misclassified as *S*; *FP*s will produce false

warnings for the classification system;

FN (false negative): The flows in *S* are misclassified as belonging to some other category; *FN*s will result in a loss of identification accuracy.

Calculation methods are as follows:

Precision: The percentage of samples classified as *S* that are truly in class *S*:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Recall: The percentage of samples in class *S* that are correctly classified as *S*:

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

Overall accuracy: The percentage of correctly classified samples:

$$Overall = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)} \tag{15}$$

4.2 Evaluation for Correction Decision of Sentiment Sentence

Accuracy, recall, and precision values serve as evaluation criteria related to both positive and negative tendencies to judge the effect of emotion key sentence judgment. Table 1 shows the label mark of the database, where A and C respectively refer to the number of positively and negatively labeled sentences. B and D respectively refer to the number of positive and negative sentences for real results. Several metrics are calculated as follows:

Po-Precision: Precision of positive sentences as expressed by

$$Po - Precision = \frac{A \bigcap B}{A} \tag{16}$$

Po-Recall: Recall of positive sentences as expressed by

$$Ne - Recall = \frac{A \bigcap B}{B} \tag{17}$$

Ne-Precision: Precision of negative sentences as expressed by

$$Ne - Precision = \frac{C \bigcap D}{C} \tag{18}$$

Ne-Recall: Recall of negative sentences as expressed by

$$Ne-Recall=\frac{C\bigcap D}{D} \tag{19}$$

Overall accuracy: The percentage of correctly classified samples:

$$Overallaccuracy=\frac{(A\bigcap B)+(C\bigcap D)}{(A+C)\bigcap(B+D)} \tag{20}$$

Table 1. Label mark of dataset

	Label mark	Real result
Positive sentence	A	B
Negative sentence	C	D

5. Experiment Results and Analysis

5.1 Chinese Opinion Analysis Evaluation (COAE) Dataset

This study used standard data from the COAE2013 dataset and label data from the COAE2014 dataset, which randomly selected 6,000 annotated sentences as training data from COAE2014; the remaining 5,000 served as test data. The corpus was training, and 1,230 positive sentiment sentences from microblogs were marked along with 1,350 negative sentiment sentences and 3,420 neutral sentiment sentences. In addition, 1,500 samples with sentiment were selected from the standard COAE2013 by annotation processing for a total training corpus from microblog of 7,500 sentences. The dataset was then divided into 10 parts.

5.2 Sentence Extraction Results

Precision (average precision), recall (average recall), and overall accuracy (average total accuracy) were used to evaluate the performance results of the 10-part dataset. The SASC spectral clustering algorithm is proposed and compared with the p1 method suggested by Jiang et al. [12] and the p2 approach put forth by Dong et al. [19]. Fig. 2 indicates that the average accuracy rate, average precision, and average recall were significantly higher in the proposed method than the other two methods; the spectral clustering model greatly reduced the time complexity, which reached an applied level. When the unlabeled dataset was larger, the other two methods did not consider optimization for the semi-supervised method, whereas the proposed method included an optimization mechanism in the semi-supervised method.

5.3 Sentence Assessment Results

Figs. 3 and 4 demonstrate that the SASC approach was the best method. A positive sentence and negative sentence were chosen to experimentally evaluate the performance of the algorithm. The COAE dataset was used as input data with a comparison to the aforementioned p1 and p2 methods. The

dataset was split into 10 parts, denoted as Experiment 1, …, 10. Let Experiment *no* be *en*, in the task; *en* varies as 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. The proposed method obtained 68% precision and 52% recall for the positive sentence in Experiment 8 and 69% precision and 51% recall for the negative sentence in Experiment 3. The SASC algorithm thus achieved higher precision and recall than p1 and p2 with different experimental data. Fig. 5 shows the overall accuracy of the SASC method to be higher compared to the other methods; in fact, the proposed approach outperformed all baseline methods because spectral clustering optimizes the k value selection of the k-means method to obtain more accurate cluster results. The overall accuracy of the SASC method was also more even, suggesting that the proposed method is more stable than others.

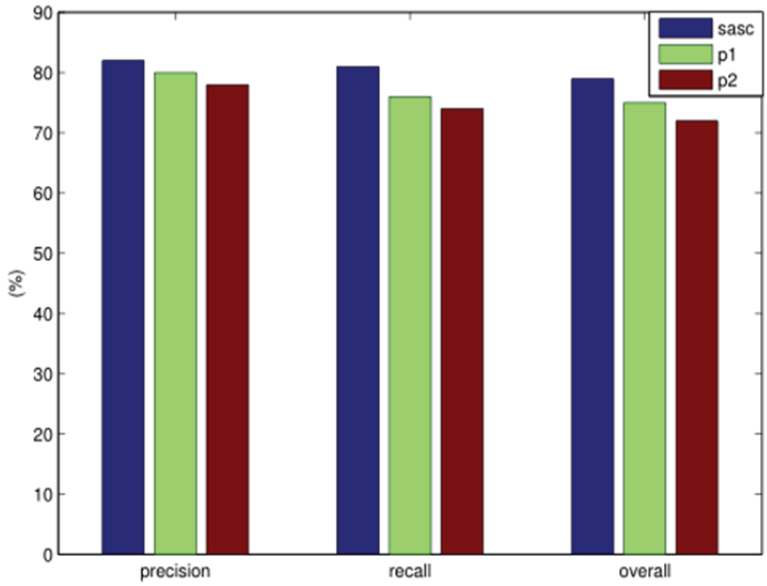
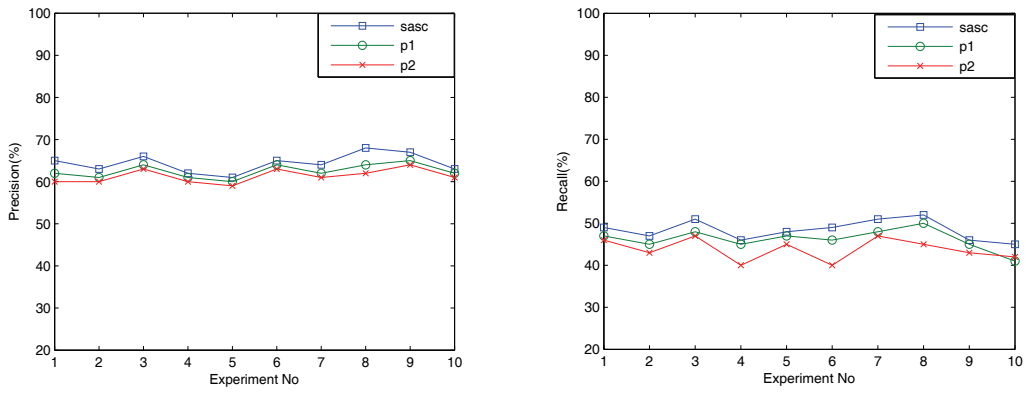


Fig. 2. Performance evaluation of sentence extraction.



(a) (b)

Fig. 3. Performance evaluation in positive sentence assessment. (a) Po-precision in positive sentence and (b) Po-recall in positive sentence.

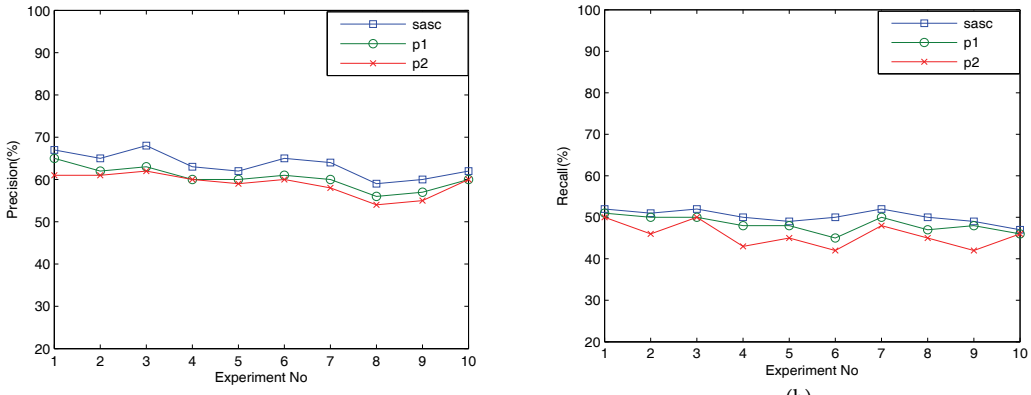


Fig. 4. Performance evaluation in negative sentence assessment. (a) Ne-precision in negative sentence and (b) Ne-recall in negative sentence.

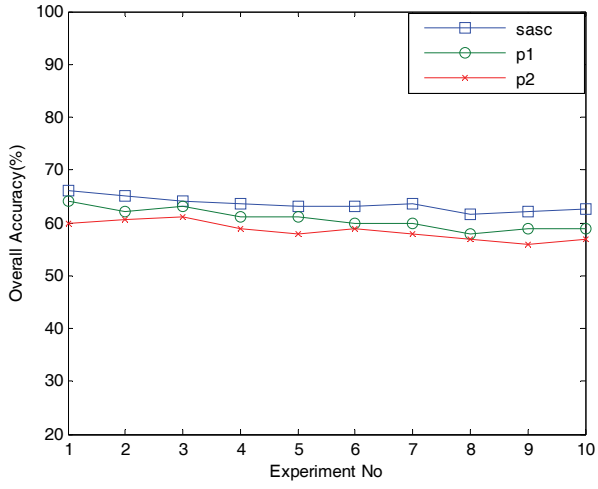


Fig. 5. Overall accuracy in sentence assessment.

6. Conclusions

To improve the accuracy of sentiment classification and solve the problem of SA on the Chinese website Weibo, this paper presents a SA model based on spectral clustering in semi-supervised ML. An optimal solution can be found through the iteration process. Experimental results show that the proposed algorithm can improve identification accuracy in Chinese microblogs without increasing the network complexity of SA. The performance of the proposed algorithm approximated that of the current traditional manual annotation algorithm. Even so, research on ML methods in SA warrants further exploration to address unresolved issues. For example, due to excessive parameters, the number of neural network models is prone to over-fitting when the model is trained. Future work will focus on reducing the complexity of structural models to identify a more suitable ML algorithm for SA.

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful comments that helped to improve the technical quality of this paper. This work was supported by the National Natural Science Foundation of China (Grant No. U1504602), China Postdoctoral Science Foundation (Grant No. 2015M572141), Science and Technology Plan Projects of Henan Province (Grant No. 162102310147), Henan Science and Technology Department of Basic and Advanced Technology Research Projects (No. 132300410276, 142300410339), and Education Department of Henan Province Science and Technology Key Project Funding (Grant No.14A520065).

References

[1] A. Balahur, R. Steinberger, M. Kabadjov, V. Zavarella, E. Van Der Goot, M. Halkia, B. Pouliquen, and J. Belyaeva, "Sentiment analysis in the news," in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010, pp. 2216-2220.

[2] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, 2011, pp. 151-160.

[3] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267-307, 2011.

[4] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: a survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093-1113, 2014.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, 2002, pp. 79-86.

[6] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 1386-1395.

[7] W. Jin, H. H. Ho, and R. K. Srihari, "OpinionMiner: a novel machine learning system for web opinion mining and extraction," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 1195-1204.

[8] J. Xu, Y. X. Ding, and X. L. Wang, "Sentiment classification for Chinese news using machine learning methods," *Journal of Chinese Information Processing*, vol. 21, no. 6, pp. 95-100, 2007.

[9] P. Yin, H. Wang, and L. Zheng, "Sentiment classification of Chinese online reviews: analysing and improving supervised machine learning," *International Journal of Web Engineering and Technology*, vol. 7, no. 4, pp. 381-398, 2012.

[10] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170-179, 2014.

[11] Y. Zhang, L. Shang, and X. Jia, "Sentiment analysis on microblogging by integrating text and image features," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Cham: Springer, 2015, pp. 52-63.

[12] F. Jiang, Y. Liu, H. Luan, M. Zhang, and S. Ma, "Microblog sentiment analysis with emoticon space model," in *Chinese National Conference on Social Media Processing*. Heidelberg: Springer, 2014, pp. 76-87.

[13] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Beijing, China, 2010, pp. 36-44.

[14] L. Pang, S. S. Li, and G. D. Zhou, "Sentiment classification method of Chinese micro-blog based on emotional knowledge," *Jisuanji Gongcheng/Computer Engineering*, vol. 38, no. 13, 2012.

[15] Q. Liu, C. Feng, and H. Huang, "Emotional tendency identification for micro-blog topics based on multiple characteristics," in *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, Bali, Indonesia, 2012, pp. 280-288.

[16] A. Go, L. Huang, and R. Bhayani, "Twitter sentiment analysis," Stanford University, *Report No. CS224N*, 2009.

[17] K. L. Liu, W. J. Li, and M. Guo, "Emoticon smoothed language models for twitter sentiment analysis," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Canada, 2012.

[18] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, "Sentence compression for aspect-based sentiment analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2111-2124, 2015.

[19] X. Dong, Q. Zou, and Y. Guan, "Set-Similarity joins based semi-supervised sentiment analysis," in *Neural Information Processing*. Heidelberg: Springer, 2012, pp. 176-183.

[20] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden, 1999, pp. 61-67.

Shi Dong https://orcid.org/0000-0003-4616-6519

He received an M.E. degree in computer application technology from the University of Electronic Science and Technology of China in 2009 and a Ph.D. in computer application technology from Southeast University in 2013. Currently, he is an associate professor at the School of Computer Science and Technology at Zhoukou Normal University and works as a post-doctoral researcher at Huazhong University of Science and Technology. He is a member of the China Computer Federation and a visiting scholar at Washington University in St. Louis. His research interests include distributed computing, network management, and evolutionary algorithms.

Xingang Zhang https://orcid.org/0000-0002-3398-5841

He received a Master's degree in computer application technology from Huazhong University of Science and Technology in 2010. Currently, he is an associate professor at the School of Computer and Information Technology at Nanyang Normal University. He is a senior member of the China Computer Federation. His research interests include distributed computing and computer networks.

Ya Li https://orcid.org/0000-0003-4839-2093

He received a B.Sc. degree in computer science and technology from Northeast Normal University and an M.Sc. degree in computer application technology from Beijing Jiaotong University, China in 1995 and 2005, respectively. He is currently conducting research on computer application technologies and mobile internet technologies.