Information Sciences 504 (2019) 61–83

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Opinion community detection and opinion leader detection based on text information and network topology in cloud environment

Chunlin Li[a,b,*], Jingpan Bai[a], Lei Zhang[b], Hengliang Tang[c], Youlong Luo[a]

[a] *Department of Computer Science, Wuhan University of Technology, Wuhan 430063, PR China*
[b] *Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, PR China*
[c] *School of Information, Beijing Wuzi University, Beijing 101149, PR China*

ARTICLE INFO

Article history:
Received 27 April 2018
Revised 27 June 2019
Accepted 30 June 2019
Available online 2 July 2019

Keywords:
Social networks
Opinion community detection
Opinion leader detection
Cloud computing

ABSTRACT

With the rapid development of web technology, the social networks have become the largest information portals. In the social platforms, the text information can effectively reflect the user opinions or the public opinions for a certain entity, such as company, celebrity service, product and so on. Therefore, mining user opinions from social networks have become an imperative requirement for the service groups. In this paper, an opinion community detection method is proposed by considering the content similarity, the time similarity and the topology structure of users. The integrated similarity between two users, which includes the content similarity, the time similarity and the topology structure of users, is achieved. Then, based on the integrated similarities, the opinion communities are detected. Furthermore, in order to identify the opinion leader, an opinion leader detection method is proposed based on the user influence and emotional analysis. The users with the same topic form the opinion community. Meanwhile, a directed graph is created to formulate the interaction relationship between users in the opinion community. Then, the user influence model and emotional analysis model are presented. Moreover, the occurrence frequencies of the negative words are also considered in the emotional analysis model. Then, a model of influence value for each user in the opinion community is built. The user with the highest influence value is considered as the opinion leader. Finally, the performances of the proposed algorithms are evaluated in a distributed computing environment. Meanwhile, the extensive experiments are conducted. The results indicate that our proposed opinion community detection algorithm can effectively detect the opinion communities. Also, the proposed opinion leader detection algorithm can significantly identify the opinion leader in the social networks.

1. Introduction

With the rapid development of web technology, the social networks have become the largest information portals. For example, Twitter, microblog, forum and BBS are the reliable, easy and quick social platform, in which the users can share anything happening around them with their friends and other followers. Every day, millions of text information are added

[*] Corresponding author at: Department of Computer Science, Wuhan University of Technology, Wuhan 430063, PR China.
E-mail addresses: chunlin74@tom.com (C. Li), lzhang@ce.ecnu.edu.cn (L. Zhang).

https://doi.org/10.1016/j.ins.2019.06.060

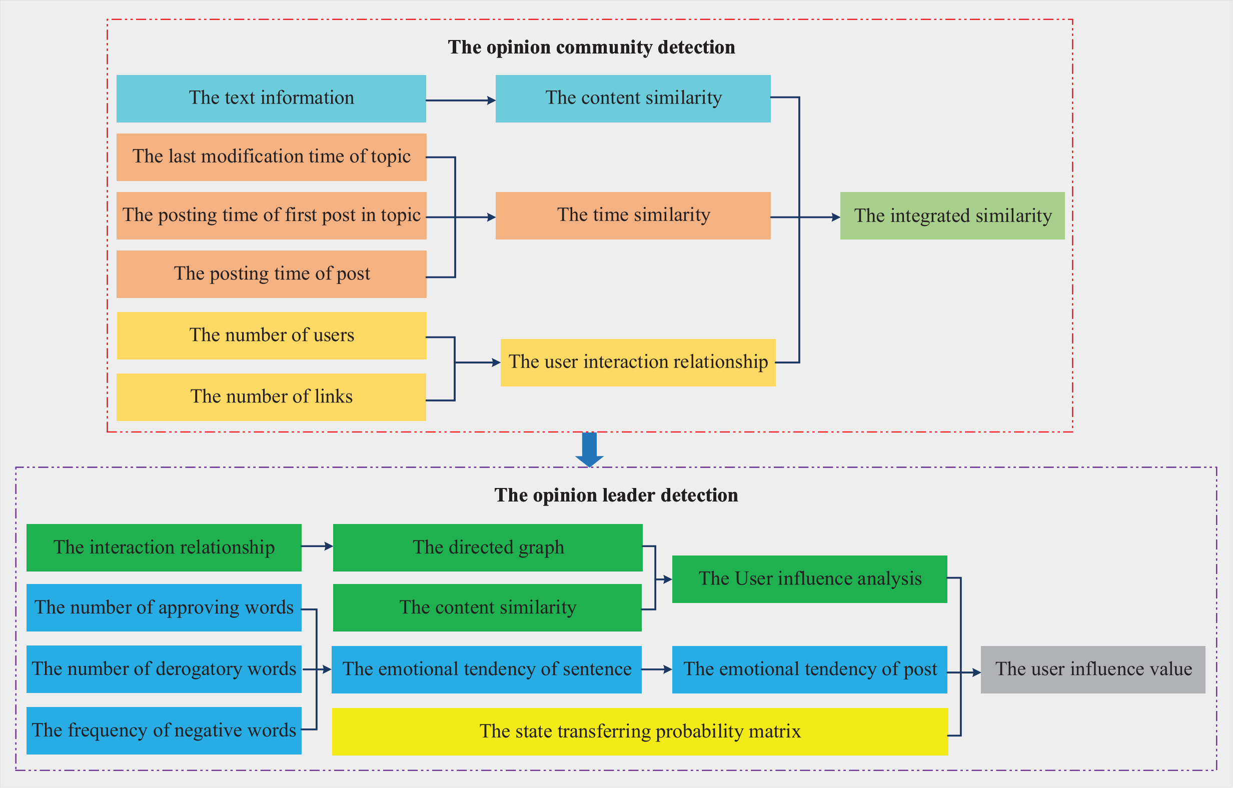
The opinion community detection
The text information
The content similarity
The last modification time of topic
The posting time of first post in topic
The time similarity
The integrated similarity
The posting time of post
The number of users
The user interaction relationship
The number of links
The opinion leader detection
The interaction relationship
The directed graph
The User influence analysis
The number of approving words
The content similarity
The number of derogatory words
The emotional tendency of sentence
The emotional tendency of post
The user influence value
The frequency of negative words
The state transferring probability matrix

Fig. 1. The overview of the proposed opinion community detection algorithm and the proposed opinion leader detection algorithm.

into the social platforms. The information, which is related to products, personal preferences, companies, politics, services, etc., makes a large part of the text information. The text information can effectively reflect the user opinions or the public opinions for a certain entity. The entity mainly includes company, celebrity service, item, product, company, public policy, etc. Especially, the companies are interested in what their customers think, thus avoiding the events potentially damaging the companys reputation in advance. Therefore, mining user opinions from social networks have become an imperative requirement for the service groups.

Some researchers addressed this problem by considering the semantic analysis or emotion analysis of the text in the social networks. However, these methods are not suitable the social networks due to the complexity of social networks. Meanwhile, it has low precision due to the monotony of the considered factor. In order to improve the performance of the opinion community detection algorithms, some scholars try to combine semantic analysis and other factors to design the opinion community detection algorithms. Also, some researchers focus on the improvement of opinion community detection method. Moreover, some scholars focus on the opinion leader detection for the analysis of online social networks. In social networks, the information is vast. The identity of each person who posts his/her opinion cannot be easily recognized. The language is not standard. The posted contents are updated in real time. These features of the forum make information acquisition difficulty. However, there exist some users who have strong professional knowledge, rich social experience and human experience. They are called as “experts” or “opinion leaders”. These experts or opinion leaders can answer some questions in time. In most cases, they are the guider of the public opinion.

In this paper, an opinion community detection method is proposed by considering the content similarity, the time similarity and the topology structure of users. Firstly, the vector space model (VSM) is introduced to mine the keywords of the text information. Also, the keyword weight computation model is built to calculate the keyword weights of the text information of posts. Secondly, the similarity models for user contents, time information and topology structure of users are introduced in detail. If two users pay attention to the same topic, the contents of their two posts have higher similarity. So, the content similarity is used to discover the users who pay attention to the same topic. By considering the posting time, the time similarity model is applied to avoid the topic drift. The topology structure is built based on the interaction behaviors between users. If there exist more interaction behaviors between two users, we think that they focus on the same topic. Finally, taking into account the content similarity, the time similarity and the topology structure of users, the integrated similarity between two users is achieved. The opinion community detection is conducted based on the integrated similarities. Fig. 1 summarizes the overview of the proposed opinion community detection algorithm and the proposed opinion leader detection algorithm.

Furthermore, in order to identify the opinion leader, a method for opinion leader detection is proposed based on the user influence and emotional analysis. Firstly, a directed graph is created to formulate the interaction relationship between users in the opinion community. Then, the user influence model is built based on the content similarity and network topology. Secondly, the emotional analysis model is presented to determine the emotional tendency of the posts. Meanwhile, the content of one post is divided into several sentences. Moreover, one sentence is expressed as a vector. The elements of the vector are the word weights. The emotional tendency of one sentence is determined based on the emotional tendencies of the words. Furthermore, the occurrence frequencies of the negative words are also considered. If the occurrence frequency of the negative words is an odd number, the emotional tendency of the sentence needs to be reversed. Otherwise, it will be kept. Finally, a model of user influence value is built based on the improved PageRank algorithm. The user with the highest influence value is considered as the opinion leader.

The main contributions of this paper are summarized as follows.

- An opinion community detection method is proposed by considering the content similarity, the time similarity and the topology structure of users. Moreover, the integrated similarity between two users is achieved based on the content similarity, the time similarity and the topology structure of users. The opinion communities are detected based on the integrated similarities.
- A method for opinion leader detection is proposed based on the user influence and emotional analysis. Furthermore, the occurrence frequencies of the negative words are also considered. The model of user influence value is built based on the improved PageRank algorithm. The user with the highest influence value is considered as the opinion leader.
- The performances of the proposed algorithms are evaluated in a distributed computing environment. The results indicate that our proposed opinion community detection algorithm can effectively detect the opinion communities. Also, the proposed opinion leader detection algorithm can significantly identify the opinion leader in social networks.

The rest of this paper is organized as follows: In Section 2, several related works are presented. In Section 3, the opinion community detection model and the opinion leader detection model are proposed, successively. In Section 4, the pseudo-code of the proposed algorithms are detailedly described. The experimental results are discussed in Section 5. Lastly, the conclusion is completed in Section 6.

2. Related work

With the development of the social networks, a lot of works have been recently conducted to study the opinion community detection problem and the opinion leader detection problem. In this section, some brief descriptions about the works are given.

2.1. Opinion community detection

Due to the usefulness of opinion community detection, many methods have been proposed to identify the opinion communities. Most researchers pay attention to the semantic analysis or emotion analysis of the text in the social networks. For example, Rill et al. [16] proposed a system named PoliTwi, which was used to detect emerging political topics in Twitter sooner than other standard information channels by extending existing knowledge bases. Rao et al. [15] presented two supervised intensive topic models to associate latent topics with emotional labels. The first model generated document-topic probability distributions by constraining topics to relevant emotions. The second model estimated word-emotion probabilities by establishing an association among terms and emotions by topics. However, the opinion community detection with semantic analysis or emotion analysis is not suitable for the complexly social networks. Meanwhile, these methods have low precision due to the monotony of the considered factor.

In order to improve the performance of the opinion community detection algorithms, some scholars try to combine semantic analysis and other factors to design the opinion community detection algorithms. For instance, Zhang et al. [26] presented a hybrid semantic analysis method to integrate semantic relations and co-occurrence relations for opinion community detection. By fusing multiple relations into a term graph and identifying topics from the graph, the hybrid semantic analysis not only identified the opinion community more effectively, but also mined the hot opinion community. Li et al. [9] studied the microblog community detection method based on the sematic analysis and the user relationships. Then, the hierarchical clustering algorithm was applied to find the communities. Yang et al. [24] proposed an incremental TF-IWF-EDF of terms part-of-speech and position weight calculation method to detect the opinion communities. The semantic and context of terms were considered and IWF was incorporated into TF-IDF. The two steps of single-pass were applied to improve the accuracy of the opinion community detection.

Also, some researchers focus on the improvement of opinion community detection method. For instance, the improvement of K-means algorithm is conducted to improve the performance of the opinion community detection. Aini et al. [2] used singular value decomposition (SVD) to reduce the large dimension of the data. The computation time required by the improved algorithm was faster than the method of K-means Clustering without any improvement. Lin et al. [13] developed an opinion community detection method based on terms to provide an unsupervised methodology for finding distinct topics from within short text system collections. Meanwhile, a K-means-based consensus clustering method was adopted to handle short text system, due to its low computational complexity and robust clustering performance.

Moreover, other methods to detect opinion community are studied to improve the accuracy of opinion community detection. For example, Xie et al. [22] designed a sketch-based opinion community detection method to leverage Twitter for early detection of bursty topics by considering the frequency of a word in twitter text and the time window size. The proposed opinion community detection method on a single machine can potentially handle hundreds of millions tweets per day which is on the same scale of the total number of daily tweets in Twitter, and present bursty events in finer-granularity. Juan et al. [6] proposed a fully unsupervised methodology to group similar content together in thematically-based topics and to organize them in the form of a concept lattice. In order to detect automatically a forum topic among the massive information, Li and Li [12] designed the opinion community detection algorithm based on agglomerative hierarchical clustering by introducing the principle of maximum entropy and information gain.

Furthermore, in order to improve the efficiency of opinion community detection algorithms, the distributed computing platforms are considered due to the efficient computation capacity [8,10,11]. For instance, Wang et al. [21] proposed a multi-layered performance analysis for big data opinion community detection applications in the cloud environment. The efficiency of opinion community detection was improved by properly configuring the cloud platform factors, such as CPU, memory, workloads, etc. Ai and Li [1] presented a parallel two-phase opinion community detection algorithm based on Apache Spark cloud computing environment. The accuracy and performance of opinion community detection were improved significantly over existing approaches.

In general, the existed methods for opinion community detection pay attention to the semantic analysis, the emotional analysis, the term, or the combination of two factors. The most of methods are the traditional methods or the improvements of the traditional methods. Of course, some scholars begin to focus on the advantages of artificial intelligence methods and distributed computing platforms. In this paper, our opinion community detection method not only considers user contents, but also focuses on the time information and the topology structure between users. Then, the integrated similarity between two users is achieved based on the content similarity, the time similarity and the topology structure of users. A two-step clustering algorithm is designed based on the single-pass algorithm and the integrated similarity. The time window is considered to satisfy the real-time of opinion community detection. Moreover, the topic strength model is built to improve the computation speed of opinion community clustering. Finally, in the experiments, the distributed computing platform is applied to improve the efficiency of opinion community detection algorithms.

2.2. Opinion leader detection

Due to the importance of opinion leader detection for the analysis of online social networks, the opinion leader detection of social networks has received more and more attention in recent years. For the opinion leader detection in social networks, most of scholars pay attention to the semantic analysis of user comments or the emotional analysis of the contents posted by users. For example, Chen et al. [3] presented a novel opinion leader detection method based on positive and negative opinions, in which the signed network was built based on online comments, meanwhile, a new model based on page trust was designed to detect the opinion leader from the opinion community. Jiang et al. [5] designed an opinion leader detection algorithm based on an improved PageRank algorithm using MapReduce. In the algorithm, the improved PageRank algorithm used the method of sentiment analysis to define the weight of the link between users and ranked users in BBS. Moreover, the proposed algorithm was conducted in the Hadoop cloud computing environment. However, the opinion leader detection with semantic analysis or emotion analysis is not suitable for the complexly social networks. Meanwhile, it has low precision due to the monotony of the considered factor.

In order to improve the performances of the opinion leader detection algorithms, some scholars try to design the opinion leader detection algorithms by considering the opinion community structure. For instance, Wang et al. [20] proposed a method of extracting the opinion leader community based on the hierarchical structure. For each post in BBS, the community components from the replies were achieved according to the structure of the community in the network. Moreover, Wang et al. [18] designed an algorithm for opinion leader detection based on the network structure and similarity of topics. In the algorithm, the links between the users were built based on user attributions, text characteristic and the topic similarities. Then, a directed-weighted network for the social network was constructed. Furthermore, the PageRank algorithm was applied to mine the opinion leaders.

Furthermore, most researchers focus on the hybrid factors for the opinion leader detection. For instance, Wang et al. [19] focused on the coupling mechanism of online public opinion events. They found that online opinion with a similar type of events had a stronger coupling, while cases with opposite psychological types also couple strongly. Ye and Du [25] proposed the opinion leader detection method based on the network topology structure, user attributions and emotional analysis. Moreover, the hierarchical structure was adopted to conduct the emotional analysis so that the malicious users, who are the possible threat to national security from the set of opinion leaders, could be identified quickly.

Also, other methods to detect opinion leaders are studied to improve the accuracy of opinion community detection. Ho et al. [4] presented an efficient approach to identify the opinion leader from group discussion without analyzing semantic and syntactic features, but costing a lot more computing effort. In the method, the degree of participation and the emotion expression from the speaking of each member during group discussion were evaluated. Meanwhile, a well-trained model for identifying opinion leaders was tested on single dataset as well as on cross dataset. Zhou et al. [27] designed the opinion leader detection algorithm by considering the characteristics of different topics and the factors that determined the

Table 1
Summary of notations.

Notations	Definition
PC_i	The vector of the text information in post i
k_{it}	The tth keyword in post i
PCW_i	The weight vector of the text information in post i
w_{is}	The weight of sth keyword
$Sim(vector_{i'}, vector_{j'})$	The similarity between vector $i^{'}$ and item $j^{'}$
P_i	Post i
C_m	Topic m
$ConSim(P_i, P_j)$	The content similarity between post i and post j
$ConSim(C_m, C_{m'})$	The content similarity between topic m and topic $m^{'}$
$ConSim(P_i, C_m)$	The content similarity between post i and topic m
$TimSim(P_i, P_j)$	The time similarity between post i and post j
$TimSim(C_m, C_{m'})$	The time similarity between topic m and topic $m^{'}$
$TimSim(P_i, C_m)$	The time similarity between post i and topic m
$TopSim(u_d, u_e)$	The interaction relationship between user d and user e
$IntSim(P_i, P_j)$	The integrated similarity between post i and post j
$IntSim(C_m, C_{m'})$	The integrated similarity between topic m and topic $m^{'}$
$IntSim(P_i, C_m)$	The integrated similarity between post i and topic m
$Str(C_m)$	The strength of topic m
$Infl_{ed}$	The influence of user e on user d
SET_x^i	The emotional tendency of sentence x in post i
PET_i	The emotional tendency of post i
Mat	The state transferring probability matrix
Por_{ed}	The state transferring probability of influence of user e on user d
PRV_d	The influence value of user d

formation and development of hot topics. Wisdom Web of Things methodology is applied to identify the new blog hot topics based on the duration, the attention degree of users, the topic novelty and the topic growth.

Usually, the existed methods for opinion leader detection pay attention to the semantic analysis, the emotional analysis, the social network topology structure, or the hybrid factors. The most of methods are the traditional methods or the improvements of the traditional methods. Of course, some scholars begin to focus on new methods from different perspectives. In this paper, our opinion leader detection method not only considers the user influence, i.e., the social network topology structure, but focus on the semantic analysis. Furthermore, the PageRank algorithm is improved to calculate the influence value of users. The user with the highest influence value is regarded as the opinion leader in the opinion community.

3. System model

In this section, an opinion community detection method is proposed by considering the content similarity, the time similarity and the topology structure of users. If two users pay attention to the same topic, the contents of their two posts have higher similarity. So, the content similarity is used to discover the users who pay attention to the same topic. Considering the posting time, the time similarity model is applied to avoid the topic drift. The topology structure is built based on the interaction behaviors between users. If there exist more interaction behaviors between two users, we think they focus on the same topic. Furthermore, in order to improve the efficiency of acquiring information, a method for opinion leader detection is proposed based on the user influence and emotional analysis. The user influence model based on the content similarity and network topology is built. The emotional analysis model is presented to determine the emotional tendency of posts, in which the content of one post is divided into several sentences. Moreover, one sentence is expressed as a vector. The emotional tendency of one sentence is determined by the emotional tendencies of the words. Furthermore, the occurrence frequencies of the negative words are also considered. If the occurrence frequency of the negative words is an odd number, the emotional tendency of the sentence needs to be reversed. Otherwise, it will be kept. Table 1 summarizes the main notations that will be used in this paper.

3.1. Opinion community detection model

In this section, an opinion community detection method is proposed by considering the content similarity, the time similarity and the topology structure of users. Firstly, the vector space model (VSM) is introduced to mine the keywords of the text information of the posts. Also, the keyword weight computation model is built to calculate the keyword weights. Secondly, the similarity models for user contents, time information and topology structure of users are introduced in detail. If two users pay attention to the same topic, the contents of their two posts have higher similarity. So, the content similarity model is used to discover the users who pay attention to the same topic. Considering the posting time, the time similarity model is built to avoid the topic drift. The topology structure is built based on the interaction behaviors between users. If there exist more interaction behaviors between two users, we think that they focus on the same topic. Finally, the integrated

Table 2
The explanation for each element of vector P_i.

Element of vector P_i	Explanation
PD_i	The ID of post i
PT_i	The title of post i
PC_i	The text information of post i
PCT_i	The posting time of post i
URL_i	The URL of post i
Au_i	The author of post i
RP_i	The list of users who respond to post i
RC_i	The list of contents that respond to post i

similarity between two users is achieved based on the content similarity, the time similarity and the topology structure of users. Furthermore, the opinion communities are detected based on the integrated similarities.

3.1.1. Text information formulation

The text information of one post can be formulated by VSM. Let $PC_i = (k_{i1}, k_{i2}, \ldots, k_{in})$ denote the vector of the text information in post i, where k_{it} denotes tth keyword in post i, and n is the number of keywords. $PCW_i = (w_{i1}, w_{i2}, \ldots, w_{in})$ represents the weight vector of the text information in post i, where w_{is} is taken as the weight of sth keyword. For convenience, we only consider two scenarios when the keyword weights are calculated. In one scenario, the keywords appear in the post title. In another scenario, the keywords appear in the content of the post. Then, the weight w_{is} is expressed by

$$w_{is} = \frac{fre_{is} e^{fre_s/Num_{kw}} I_{is}}{Num_{post} \sqrt{\sum_{j=1}^{Num_{kw}^{j}} \left(fre_{js}\right)^2}}, \tag{1}$$

where Num_{post} is taken as the number of posts. Num_{kw}^{j} denotes the number of keywords in post j. fre_{js} represents the frequency of keyword s in post j. fre_s is taken as the frequency of keyword s in all posts. Num_{kw} is denoted as the total number of keywords in all posts. $I_{is} = 1$ if keyword s appears in the content of post i. $I_{is} = 1 + \varepsilon$ $(\varepsilon > 0)$ if the title of post i includes keyword s.

Given $vector_{i'} = (v_{i'1}, v_{i'2}, \ldots, v_{i'n'})$ and $vector_{j'} = (v_{j'1}, v_{j'2}, \ldots, v_{j'n'})$, the similarity between two vectors can be expressed by

$$Sim(vector_{i'}, vector_{j'}) = \frac{\sum_{s'=1}^{n'} v_{i's'} \sum_{s'=1}^{n'} v_{j's'}}{\sqrt{\sum_{s'=1}^{n'} (v_{i's'})^2 \sum_{s'=1}^{n'} \left(v_{j's'}\right)^2}}, \tag{2}$$

where n' is the size of the vector.

3.1.2. Integrated similarity

The opinion community detection is closely related to text information, posting time and interaction behavior between users. If two users pay attention to the same topic, the contents of their two posts have higher similarity. So, the content similarity is used to discover the users who pay attention to the same topic. Like the development of the news event, the development of one post needs to go through four phases of latency, growth, maturation and decline. Therefore, two posts are regarded to describe the same topic if they have similar posting time. The metric of posting time can also avoid the topic drift between the contents with the same topic. In the forum, one user mainly has two ways to express his/her interests in one certain topic. One is to browse the post within the topic. Another one is to discuss the topic by posting his/her own opinion. Due to the invisibility of the browsing behavior, we only consider the discussion behavior between users in this paper. If one user responds to another user by posting his/her own opinion, a link between the two users is built. The number of links between two users is greater, the relationship between them is closer. In other words, two users with more links have more possible to pay attention to the same topic.

Post i can be formulated into a vector as follows:

$$P_i = (PD_i, PT_i, PC_i, PCT_i, URL_i, Au_i, RP_i, RC_i), \tag{3}$$

where the explanation for each element of vector P_i is shown in Table 2.

The text information of one post can be formulated by VSM. The posting time of posts is used to value the relationship between posts. The interval time between posting time of two posts is greater, the possibility that two posts pay attention to the same topic is smaller. The URL of one post is mainly used to quickly find the source of the post. The list of users and the list of contents are used to discover the opinion leader.

Topic m can also be expressed by a vector as follows:

$$C_m = (CD_m, CKW_m, CCT_m, LPD_m, CLMT_m), \tag{4}$$

where the explanation for each element of vector C_m is shown in Table 3.

Table 3
The explanation for each element of vector C_m.

Element of vector C_m	Explanation
CD_m	The ID of topic m
CKW_m	The keywords that express topic m
CCT_m	The creation time of topic m
LPD_m	The list of posts in topic m
$CLMT_m$	The last modification time of topic m

For convenience, the top ten keywords with most weight are selected to express one topic. The creation time and last modification time of one topic are used to value the time relationship between two topics or between one topic and one post. The text information of one post can be formulated by VSM. The content similarity is used to discover the users who pay attention to the same topic. Let $ConSim(P_i, P_j)$ denote the content similarity, which can be expressed by

$$ConSim(P_i, P_j) = Sim(PCW_i, PCW_j), \tag{5}$$

where P_i and P_j denote the post i and post j, respectively. PCW_i and PCW_j represent the weight vector of the text information in post i and post j, respectively. $Sim(PCW_i, PCW_j)$ is taken as the similarity between vector PCW_i and PCW_j, which is calculated by Eq. (2).

Let CC_m be the keyword vector of topic m. CCW_m is defined as the corresponding weight vector of the keywords of topic m. The top ten keywords with most weight are selected to form the keyword vector of topic m. CCW_m is calculated by Eq. (1). Then, the content similarity between topic m and topic m' is

$$ConSim(C_m, C_{m'}) = Sim(CCW_m, CCW_{m'}), \tag{6}$$

where $CCW_{m'}$ is the weight vector of the keywords of topic m'.

Also, the content similarity between post i and topic m is

$$ConSim(P_i, C_m) = Sim(PCW_i, CCW_m). \tag{7}$$

Like the development of the news event, the development of one post needs to go through four phases of latency, growth, maturation and decline. Therefore, two posts are regarded to describe the same topic if they have similar posting time, which can also avoid the topic drift between the contents with the same topic. Let TCF_m denote the posting time of the first post in topic m. TCL_m is taken as the posting time of the last post in topic m. In other words, TCL_m is the last modification time of topic m, i.e., $TCL_m = CLMT_m$. Then, the creation time of topic m is expressed by

$$CCT_m = \sqrt{\frac{(TCF_m)^2 + (TCL_m)^2}{2}}. \tag{8}$$

Let $TimSim(P_i, C_m)$ denote the time similarity between post i and topic m. It can be expressed by

$$TimSim(P_i, C_m) = PCT_i - CCT_m, \tag{9}$$

where PCT_i is the posting time of post i. Similarly, the time similarity between topic m and topic m' is

$$TimSim(C_m, C_{m'}) = |CCT_m - CCT_{m'}|. \tag{10}$$

The time similarity between post i and post j is

$$TimSim(P_i, P_j) = PCT_i - PCT_j. \tag{11}$$

In the forum, one user mainly has two ways to express his/her interests in one certain topic. One is to browse the post within the topic. Another one is to discuss the topic by posting his/her own opinion. Due to the invisibility of the browsing behavior, we only consider the discussion behavior between users in this paper. If one user responds to another user by posting his/her own opinion, a link between the two users is built. The number of links between two users is greater, the relationship between them is closer. In other words, two users with more links have more possible to pay attention to the same topic. Let $TopSim(u_d, u_e)$ denote the interaction relationship between user d and user e. It is defined by

$$TopSim(u_d, u_e) = \frac{l_{de}}{\sum_{e=1}^{U} \sum_{d=1}^{U} l_{de}}, \tag{12}$$

where U denotes the number of users, and l_{de} is taken as the number of links. If one user responds to another user by posting his/her own opinion, a link between the two users is built. $\sum_{e=1}^{U} \sum_{d=1}^{U} l_{de}$ is the total number of links between all users. Obviously, the user will not respond to his/her own post. If one user does so, the number of links between his/her own is defined as zero, i.e., $l_{dd} = 0$.

Then, the integrated similarity between post i and post j is expressed by

$$IntSim(P_i, P_j) = \alpha ConSim(P_i, P_j) + \beta TimSim(P_i, P_j) + \gamma TopSim(u_d, u_e), \tag{13}$$

where α, β and γ are the weight coefficients, and $\alpha + \beta + \gamma = 1$.

The integrated similarity between post i and topic m takes the form of

$$IntSim(P_i, C_m) = \alpha' ConSim(P_i, C_m) + \beta' TimSim(P_i, C_m) + \gamma' \sum_{d=1}^{Num_m^C} TopSim(u^i, u_d), \tag{14}$$

where $\alpha^{'}$, $\beta^{'}$ and $\gamma^{'}$ are the weight coefficients, and $\alpha^{'} + \beta^{'} + \gamma^{'} = 1$. u^i is the user who publishes post i. Num_m^C denotes the number of users who publish the posts that are related to topic m.

The integrated similarity between topic m and topic $m^{'}$ is

$$IntSim(C_m, C_{m'}) = \alpha'' ConSim(C_m, C_{m'}) + \beta'' TimSim(C_m, C_{m'}) + \gamma'' \sum_{d=1}^{Num_m^C} \sum_{e=1}^{Num_{m'}^C} TopSim(u_d, u_e), \tag{15}$$

where $\alpha^{''}$, $\beta^{''}$ and $\gamma^{''}$ are the weight coefficients, and $\alpha^{''} + \beta^{''} + \gamma^{''} = 1$. $Num_{m'}^C$ is taken as the number of users who publish posts that are related to topic m'.

3.1.3. Opinion community updating

In order to find the opinion community, the single-pass algorithm is taken as the clustering algorithm to discover the communities. Single-pass algorithm is an incremental clustering algorithm. It has low time complexity so that it is widely used in the topic clustering, community detection, etc. [7]. Then, a two-step clustering algorithm is designed based on the single-pass algorithm and the integrated similarity.

Firstly, the existed posts are clustered by the single-pass algorithm, and the classification of posts is achieved. During a certain period, such as one day, one hour, etc., the new posts will be obtained by the crawler algorithm. In the first step, the new posts are clustered based on the single-pass algorithm and the integrated similarity. In the second step, the integrated similarity between each cluster of new posts and each cluster of old posts is calculated. If the integrated similarity between one cluster of new posts and one cluster of old posts is more than the given threshold, these two clusters are merged into one new cluster. Otherwise, the cluster of new posts will be taken as a new cluster and participate in the group of existed clusters.

However, as time goes by, the number of clusters becomes large. Meanwhile, the required storage space for storing the posts become large. Moreover, the clustering computation time becomes large. Also, the hot of one topic is closely related to time. As time goes by, it decreases and disappeared at last. So, in order to save the storage space and improve the computation speed of the post clustering, the topic strength model is proposed to choose the topics with little hot and then delete them. Let $Str(C_m)$ denote the strength of topic m. $Str(C_m)$ is expressed by

$$Str(C_m) = Num_m^{post} / Num_{post}^{new}, \tag{16}$$

where Num_{post}^{new} is taken as the number of new posts, and Num_m^{post} is the number of posts including topic m. Given a threshold of topic strength, if the strength of one topic is less than the threshold for seven periods, this topic will be deleted.

3.2. Opinion leader detection

In the forum, the information is vast. The identity of each person who publishes his/her opinion cannot be easily recognized. The language is not standard. Meanwhile, the contents of posts are updated in real time. These forum features make information acquisition difficulty. However, in the forum, there exist some users who have strong professional knowledge, rich social experience and human experience. They are called "experts" or "opinion leader". They can answer some questions in time. So, with the help of "opinion leader", users can acquire the information that they want.

In this section, a method for opinion leader detection is proposed based on the user influence and emotional analysis. Firstly, a directed graph is created to formulate the interaction relationship between users in the opinion community. Then, the user influence model is built based on the content similarity and network topology. Secondly, the emotional analysis model is presented to determine the emotional tendency of posts, in which the content of one post is divided into several sentences. Moreover, one sentence is expressed as a vector. The emotional tendency of one sentence is determined by the emotional tendencies of the words. Furthermore, the occurrence frequencies of the negative words are also considered. If the occurrence frequency of the negative words is an odd number, the emotional tendency of the sentence needs to be reversed. Otherwise, it will be kept. Finally, the user influence value is achieved based on the improved PageRank algorithm. The user with the highest influence value is considered as the opinion leader in the opinion community.

3.2.1. User influence analysis

In the forum, one user expresses his/her opinion or idea by publishing posts. Other users discuss the opinion with the user by responding to their interesting posts. One user publishes his/her opinion, other users reply to the post, and the user replies to them again. After circular interaction for many times, one user will influence other users in an invisible and indirect manner. Obviously, one user with more communications is much more possible to be the opinion leader. To find the opinion leader, a directed graph is used to formulate the relationship between users. The rules for creating the directed graph is described as follows:

(1) One user is taken as one node;

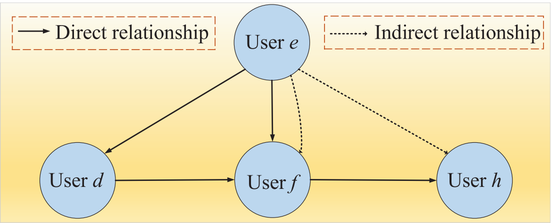

Fig. 2. The relationship between users.

Table 4
The example of emotional tendencies of the words.

The contents of one post in the forum	The emotional tendencies
Strongly support the guidelines of the party central committee.	Positive
I keep my opinion on what you said.	Neutral
This watch is not as good as the post says.	Negative

Table 5
The part of the benchmark set.

The benchmark for approving words			The benchmark for derogatory words		
Right	Successful	Easy	Wrong	Failing	Difficult
Real	Like	Smart	Sham	Dislike	Foolish
Positive	Advanced	Famous	Negative	Backward	Common
Maturity	High quality	Pace	Childish	Inferior	Turbulent
Perfect	Consummate	Noble	Faulty	Flaw	Adulterate
Beautiful	Supernormal	Enlightened	Ugly	Error	Behindhand
Tender	Encourage	Thrifty	Explosive	Crackdown	Waste

(2) If user d replies to user e by posting his/her opinion, a directed and solid line from user e to user d is built. Furthermore, if user f replies to user d, there exist not only a directed and solid line from user d to user f, but also a directed and dotted line from user e to user f.

For example, user e publishes one post. User d replies to user e. User f replies to user d and user e. User h replies to user f. Then, the relationship graph between them is built in Fig. 2. User e not only directly influences user f, but has an indirect influence on user f. Meanwhile, user e has an indirect influence on user h.

Obviously, the influence of one user on other users is determined based on not only the posting contents, but also the posting order. Let $Infl_{ed}$ denote the influence of user e on user d. $Infl_{ed}$ is expressed by

$$Infl_{ed} = \lambda ConSim(u_e, u_d) + (1-\lambda)(DH(e,d) + IDH(e,d)), \tag{17}$$

where λ is the weight coefficient, and $0 \le \lambda \le 1$. $ConSim(u_e,\ u_d)$ denotes the posting content similarity between user e and user d. $DH(e,\ d)$ and $IDH(e,\ d)$ are taken as the direct relationship and indirect relationship between user e and user d, respectively. $DH(e,d) = \sigma^{|Dis(e,d)|}$, where $Dis(e,\ d)$ is the directly logical distance between user e and user d. $IDH(e,d) = \sigma^{|Dis'(e,d)|}$, where $Dis'(e,\ d)$ denotes the indirectly logical distance between user e and user d. For instance, in Fig. 2, $Infl_{ed} = \lambda ConSim(u_e, u_d) + (1-\lambda)\sigma$, $Infl_{ef} = \lambda ConSim(u_e, u_f) + (1-\lambda)(\sigma + \sigma^2)$, and $Infl_{eh} = (1-\lambda)\sigma^3 + \lambda ConSim(u_e, u_h)$.

3.2.2. Emotional polarity analysis

The emotional polarity analysis of the words is conducted based on a corpus or dictionary. In this paper, the emotional tendencies of Chinese words are judged based on the How Net emotion dictionary. Actually, the emotional tendencies of words mainly include positive, neutral and negative. Table 4 shows the emotional tendency judgment of the words.

The main method to judge the emotional tendencies of words is described as follows:

Firstly, some words are selected from the How Net emotion dictionary, and they form a benchmark set. The set includes both the approving words and the derogatory words. Table 5 shows the part of the benchmark set.

To determine the emotional tendency of one word, we need to traverse the set and find the occurrence frequency of the word in the approving word set and the derogatory word set, respectively. Then, the emotional tendency of one sentence is determined based on the occurrence frequency of the keywords in the sentence. Post i is divided into g sentences based on the punctuation. Then, the ICTCLAS [28] (Institute of Computing Technology, Chinese Lexical Analysis System) is used to split in z words. So, in post i, xth sentence is formulated as a set $Sen_x^i = \{Word_{x1}^i, Word_{x2}^i, \ldots, Word_{xz}^i\}$, where $x = 1, 2, \ldots, g$. Let App_x^i denote the number of approving words in the sentence x of post i. Der_x^i is used to define the number of derogatory words in the sentence x of post i. Then, the emotional tendency of sentence x in post i is

$$SET_x^i = \frac{(App_x^i - Der_x^i)}{z} \times \Gamma_x^i, \tag{18}$$

where SET_x^i is taken as the emotional tendency of sentence x in post i, and Γ_x^i is an indicator function. $\Gamma_x^i = -1$ if the occurrence frequency of the negative words is odd number. Otherwise, $\Gamma_x^i = 1$. Obviously, $SET_x^i > 0$ means that the emotional tendency of sentence x in post i is approving. $SET_x^i < 0$ implies that the emotional tendency of sentence x in post i is derogatory. $SET_x^i = 0$ denotes that the emotional tendency of sentence x in post i is neutral.

Furthermore, the emotional tendency of post i is expressed as

$$PET_i = \sum_{x=1}^{g} \Phi_x^i, \tag{19}$$

where PET_i denotes the emotional tendency of post i, and Φ_x^i is an indicator function. $\Phi_x^i = 1$ if $SET_x^i > 0$. $\Phi_x^i = -1$ if $SET_x^i < 0$. $\Phi_x^i = 0$ if $SET_x^i = 0$.

3.2.3. Opinion leader detection

In this section, the PageRank algorithm is improved to suit for the opinion leader detection in the forum. The method for opinion leader detection is described in detailed as follows. The total influence value of users in one certain opinion community is set to 1. Initially, the total influence value is allocated to one user in the opinion community. Then, the influence will be transmitted to other users through the social network in the opinion community. Until the transition probability of the influence value is balanced gradually, each user has an influence value. Finally, the user with the highest influence value is considered as the opinion leader.

In the process of opinion leader detection, the state transferring probability matrix is defined as

$$Mat = (Por_{ed})_{q \times q}, \tag{20}$$

where q denotes the number of users in the opinion community, and Por_{ed} is the state transferring probability of the influence of user e on user d.

$$Por_{ed} = \frac{Infl_{ed}}{\sum_{e=1}^{q} \sum_{d \neq e}^{q} Infl_{ed}} \times \frac{1}{STW_e}, \tag{21}$$

where $Infl_{ed}$ denotes the influence of user e on user d. $Infl_{ed} / \sum_{e=1}^{q} \sum_{d \neq e}^{q} Infl_{ed}$ is taken as the standardization of $Infl_{ed}$. $1/STW_e$ is defined as the state transferring weight that quantizes the influence transmission from user e to other suer. STW_e is the number of links from user e to other users.

The influence value of user d is expressed by

$$PRV_d = \frac{1-\delta}{q} + \delta \sum_{e \in \Delta_d} \left(\frac{PRV_e}{STW_e} \times Infl_{ed} \times PET_i^e \right), \tag{22}$$

where PRV_d denotes the influence value of user d. q is defined as the number of users in the opinion community. User e replies to user d by publishing post i. δ is the probability that the content of post i is related to the post of user d. In general, $\delta = 0.85$. Δ_d is the set of users who have a link from themselves to user d. STW_e is the number of links from user e to other users. $Infl_{ed}$ denotes the influence of user e on user d. PET_i^e is taken as the emotional tendency of post i published by user e for replying to user d, which can be calculated by Eq. (19).

4. Algorithm

In this section, the opinion community detection algorithm is presented based on the content similarity, the time similarity and the topology structure of users. Furthermore, the opinion leader discovery algorithm is proposed based on the user influence and emotional analysis. The detailed description of the two algorithms is shown as follows.

4.1. The opinion community detection algorithm based on the content similarity, the time similarity and the topology structure of users

Algorithm 1 represents the pseudo-code of the opinion community detection algorithm. The input information can be obtained by the web crawler algorithm. When the new posts are obtained, the integrated similarities between posts are calculated based on the content similarity, the time similarity and the topology structure of users (Algorithm 1 line 1–11). The new topic set is achieved by clustering the new posts with the help of Single-Pass algorithm (Algorithm 1 line 12). Similarly, the integrated similarities between the topics in new topic set and the topics in old topic set are calculated (Algorithm 1 line 16–23). Then, the final topic set is achieved by clustering the new and old topics with the help of Single-Pass algorithm, i.e., the existed topic set is updated (Algorithm 1 line 24). Finally, the strength of each topic is calculated (Algorithm 1 line 26). If the strength of one topic is less than the threshold for seven periods, this topic will be deleted to

Algorithm 1 The opinion community detection algorithm based on the content similarity, the time similarity and the topology structure of users.

```
Input: The new post set: $P = \{P_1, P_2, \ldots, P_{NP}\}$.
       The posting time set of the posts: $PT = \{PT_1, PT_2, \ldots, PT_{NP}\}$.
       The number of links between users.
       The topic set: $C = \{C_1, C_2, \ldots, C_{NC}\}$.
       The set of loop count variables for topics: $Count = \{Cou_1, Cou_2, \ldots, Cou_{NC}\}$.
       The posting time set of the last post in topics: $TCL = \{TCL_1, TCL_2, \ldots, TCL_{NC}\}$.
       The posting time set of the first post in topics: $TCF = \{TCF_1, TCF_2, \ldots, TCF_{NC}\}$.
Output: The new topic set: $C' = \{C_1^{'}, C_2^{'}, \ldots, C_{NC'}^{'}\}$.

for each post $P_i \in P$ do
    Calculate $PCW_i$ // The weight vector of the text information in post $i$
end for
for each post $P_i \in P$ do
    for each post $P_i \in P(j > i)$ do
        Calculate $ConSim(P_i, P_j)$ // The content similarity between post $i$ and post $j$
        Calculate $TimSim(P_i, P_j)$ // The time similarity between post $i$ and post $j$
        Calculate $TopSim(u_d, u_e)$ // The interaction relationship between user $d$ and user $e$
        Calculate $IntSim(P_i, P_j)$ // The integrated similarity between post $i$ and post $j$
    end for
end for
The new topic set $C^{new} = \{C_1^{new}, C_2^{new}, \ldots, C_{NC^{new}}^{new}\}$ is achieved by clustering the new posts based on the integrated similarity and Single-Pass algorithm
if $C = \phi$ then
    The initial value of the loop count variable for each topic is set to 7
else
    for each topic $C_m \in C$ do
        for each topic $C_{m'}^{new} \in C^{new}$ do
            Calculate $ConSim(C_m, C_{m'}^{new})$ // The content similarity between topic $m$ and post $m^{'}$
            Calculate $TimSim(C_m, C_{m'}^{new})$ // The time similarity between topic $m$ and post $m^{'}$
            Calculate $TopSim(C_m, C_{m'}^{new})$ // The interaction relationship between topic $m$ and post $m^{'}$
            Calculate $IntSim(C_m, C_{m'}^{new})$ // The integrated similarity between topic $m$ and post $m^{'}$
        end for
    end for
    The final topic set $C^{'} = \{C_1^{'}, C_2^{'}, \ldots, C_{NC^{'}}^{'}\}$ is achieved by clustering the new posts based on the integrated similarity and Single-Pass algorithm
    for each topic $C_{m''}^{'} \in C'$ do
        Calculate $Str(C_{m''}^{'})$ // The strength of topic $m''$ in the final topic set $C'$
        if topic $C_{m''}^{'}$ is a new topic then
            $Cou_{m''}^{'} = 7$
        else
            if $Str(C_{m''}^{'}) < threshold$ then
                $Cou_{m''}^{'} - -$
                if $Cou_{m''}^{'} == 0$ then
                    Remove topic $C_{m''}^{'}$ from the final topic set $C'$
                end if
            end if
        end if
    end for
end if
$C \leftarrow C'$ // Update the final topic set $C'$ as the topic set $C$
Update the set of loop count variables for the corresponding topics
```

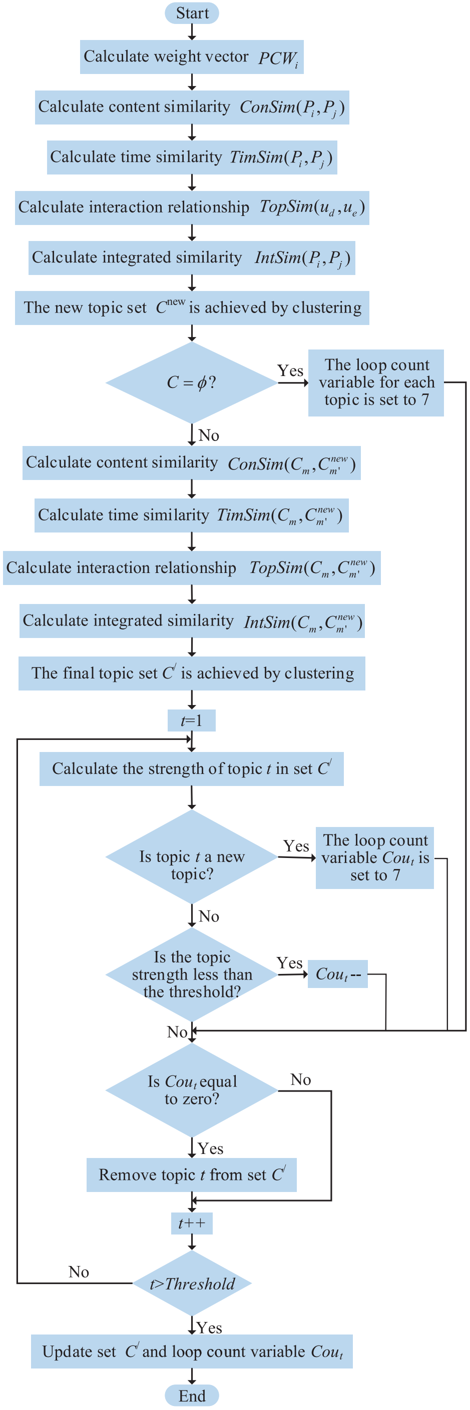
Start
Calculate weight vector PCW_i
Calculate content similarity $ConSim(P_i, P_j)$
Calculate time similarity $TimSim(P_i, P_j)$
Calculate interaction relationship $TopSim(u_d, u_e)$
Calculate integrated similarity $IntSim(P_i, P_j)$
The new topic set C^{new} is achieved by clustering
$C = \phi$?
Yes
The loop count variable for each topic is set to 7
No
Calculate content similarity $ConSim(C_m, C_{m'}^{new})$
Calculate time similarity $TimSim(C_m, C_{m'}^{new})$
Calculate interaction relationship $TopSim(C_m, C_{m'}^{new})$
Calculate integrated similarity $IntSim(C_m, C_{m'}^{new})$
The final topic set C' is achieved by clustering
t=1
Calculate the strength of topic t in set C'
Is topic t a new topic?
Yes
The loop count variable Cou_t is set to 7
No
Is the topic strength less than the threshold?
Yes
Cou_t--
No
Is Cou_t equal to zero?
No
Yes
Remove topic t from set C'
t++
No
t>Threshold
Yes
Update set C' and loop count variable Cou_t
End

Fig. 3. The flow chart of Algorithm 1.

save the storage space and improve the computation speed of post clustering (Algorithm 1 line 27–38). The flow chart of algorithm 1 is shown in Fig. 3.

The time complexity of the opinion community detection algorithm mainly consists of the time complexity of new post clustering and the time complexity of the clustering of new and old topics. The time complexity of new post clustering is $O(m^2)$, where m is the number of new posts. The time complexity of the clustering of new and old topics is $O(kn)$, where

k is the number of topics for the new posts, and n denotes the number of existed topics. Therefore, the time complexity of the opinion community detection algorithm is $O(m^2 + kn)$.

4.2. The opinion leader discovery algorithm based on the user influence and emotional analysis

Algorithm 2 depicts the pseudo-code of the opinion leader discovery algorithm based on the user influence and emotional analysis. The opinion community is mined in Section 3.1. Firstly, the topology structure of the opinion community is built (Algorithm 2 line 1). Then, for each user in the opinion community, the posting content similarity, the direct relationship, the indirect relationship, the emotional tendency and the state transferring probability matrix are computed, successively (Algorithm 2 line 2–15). Furthermore, the total influence value of users in one opinion community is set to 1. The total influence value is allocated to one user in the opinion community (Algorithm 2 line 16). Then, the influence will be transmitted to other users through the social network in the opinion community, and the influence value of each user in the opinion community is calculated based on the improved PageRank algorithm with Eq. (22). Until the transition probability of the influence is balanced gradually, each user in the opinion community has an influence value. Finally, the list of users with influence value is achieved, and the user with the highest influence value is considered as the opinion leader (Algorithm 2 line 16). The flow chart of Algorithm 2 is shown in Fig. 4.

Algorithm 2 The opinion leader discovery algorithm based on the user influence and emotional analysis.

```
Input: The post set in one opinion community: $P = \{P_1, P_2, \ldots, P_{NP}\}$.
       The user set in the opinion community: $U = \{u_1, u_2, \ldots, u_n\}$.
Output: The list of users with influence value: $UL = \{u_1, v_1; u_2, v_2; \ldots; u_n, v_n\}$.
       The opinion leader in the opinion community.
1: Build the topology structure of the opinion community
2: for each user $u_i \in U$ do
3:     for each user $u_j \in U$ $(j \neq i)$ do
4:         Calculate $ConSim(u_i, u_j)$ // The posting content similarity between user $i$ and user $j$
5:         Calculate $DH(i, j)$ // The direct relationship between user $i$ and user $j$
6:         Calculate $IDH(i, j)$ // The indirect relationship between user $i$ and user $j$
7:         Calculate $Infl_{ij}$ // The influence of user $i$ on user $j$
8:         for each sentence $x$ in post $P_k$ published by user $u_j$ do
9:             Calculate $SET_x^i$ // The emotional tendency of sentence $x$ in post $P_k$
10:        end for
11:        Calculate $PET_j$ // The emotional tendency of post $P_k$ published by user $u_j$
12:        Calculate $Por_{ed}$ // The state transferring probability of the influence of user $e$ on user $d$
13:        Calculate $Mat$ // The state transferring probability matrix
14:    end for
15: end for
16: $UL = \{u_1, 1; u_2, 0; \ldots; u_n, 0\}$ // The list of users with influence value is initialized
17: The influence value of each user is calculated based on the improved PageRank algorithm with Eq. (22)
18: Update the list of users with influence value
19: The user with the highest influence value is considered as the opinion leader, and the opinion leader is achieved
```

The time complexity of the opinion leader discovery algorithm mainly consists of the time complexity of the algorithm for calculating user influence and emotional polarity and the time complexity of the opinion leader discovery algorithm. The time complexity of the algorithm for calculating user influence and emotional polarity is $O(n^2 ms)$, where n is the number of users in the opinion community. $m = \max\{p_i; i = 1, 2, \ldots, n\}$, where p_i denotes the number of posts that are related to the opinion community and are published by user i. $s = \max\{q_{ij}; i = 1, 2, \ldots, n, j = 1, 2, \ldots, p_i\}$, where q_{ij} is taken as the number of sentences in post j published by user i. The time complexity of the opinion leader discovery algorithm is $O(n^2)$, where n is the number of users in the opinion community. Therefore, the time complexity of the opinion leader discovery algorithm is $O(n^2 ms)$. In this paper, the parallel-distributed computing environment is used to run the proposed algorithm, then, the time complexity of the opinion leader discovery algorithm is reduced to $O(n^2 ms/lk)$, where l is the number of nodes in the parallel-distributed computing environment, and k is the number of tasks that are processed by one node.

5. Experimental results and analysis

5.1. Experiment environment

In this section, the opinion community detection algorithm and the opinion leader discovery algorithm would be experimentally verified. In the experiments, the Hadoop 0.20.2, which is formed by 3 computers, is used to conduct the experi-

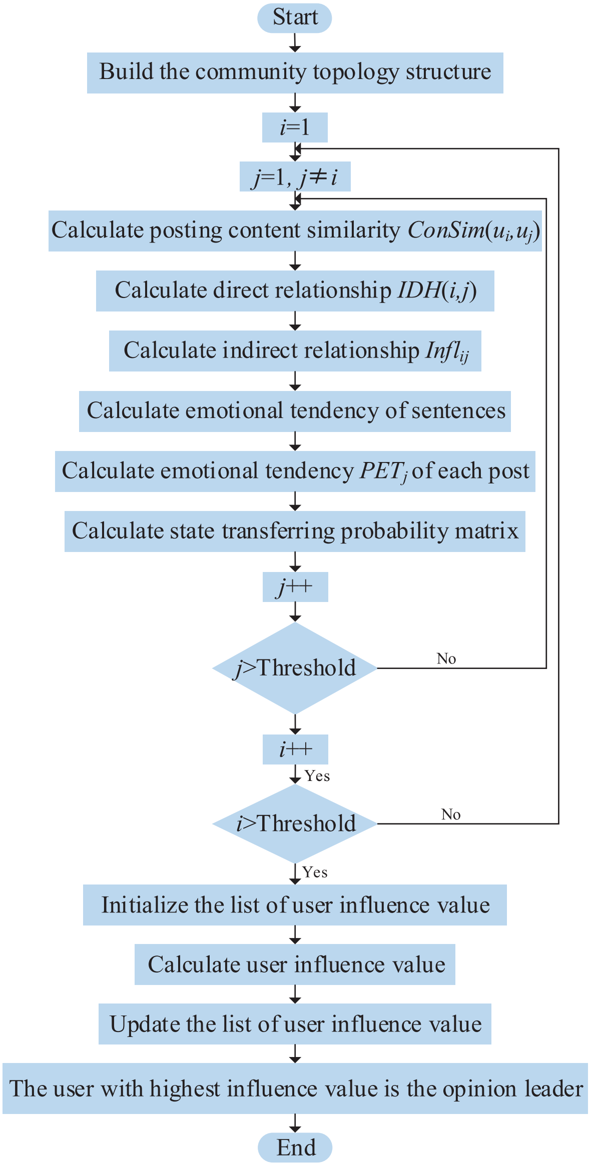
Start
Build the community topology structure
i=1
j=1, j≠i
Calculate posting content similarity ConSim(u_i,u_j)
Calculate direct relationship IDH(i,j)
Calculate indirect relationship Infl_ij
Calculate emotional tendency of sentences
Calculate emotional tendency PET_j of each post
Calculate state transferring probability matrix
j++
j>Threshold
No
i++
Yes
i>Threshold
No
Yes
Initialize the list of user influence value
Calculate user influence value
Update the list of user influence value
The user with highest influence value is the opinion leader
End

Fig. 4. The flow chart of Algorithm 2.

ments. One computer is taken as the master and it has Intel 2.5 GHz Core i5 CPU, 4 GB RAM and 500 GB HDD. Other two computers are regarded as slaves, and each one of them has Intel 3.3 GHz Core i3, 8 GB RAM and 1T HDD. Ubuntu 14.04 is taken as the operating system to run on each computer. The version of the Java Development Kit is jdk-7-linux-i586.tar.gz.

5.2. Data set

In the experiments, all data are achieved from the end of the world forum [23]. The web crawler [14] is used to achieve the text information of each user in the forum, such as user ID, the set of posts, the time of posting time, etc. 11.713 thousand posts are crawled and 8976 topics are achieved.

5.3. Benchmark algorithms

In order to evaluate the performance of the opinion community detection (OCD) algorithm based on the content similarity, the time similarity and the topology structure of users. The single-pass (SP) algorithm and the *k*-means (KM) algorithm [17] are taken as the benchmark algorithms. As the traditional clustering algorithm, the KM algorithm needs a subjective

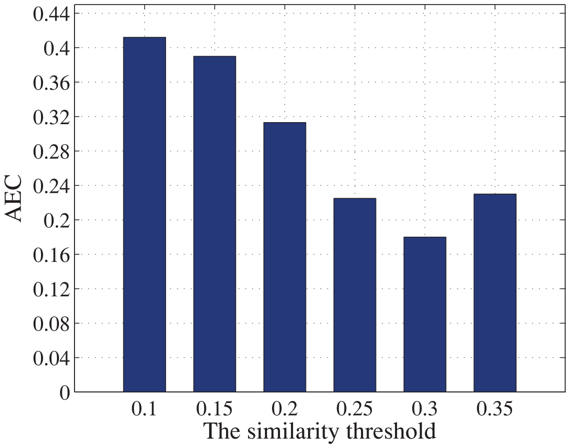

Fig. 5. The value of AEC with different similarity thresholds.

threshold to determine the number of clusters. Generally, the threshold is set by experts and experienced people. SP algorithm is the foundation of OCD algorithm. The OCD algorithm is designed to reduce the computation time and improve the efficiency of data process. Therefore, it is reasonable that the KM algorithm and the SP algorithm are taken as the benchmark algorithms to value the performance of OCD algorithm.

In order to value the performance of the opinion leader discovery algorithm based on the user influence and emotional analysis (OLDUIEA), the online-time-based opinion leader discovery (OTOLD) algorithm, the experience-based opinion leader discovery (EOLD) algorithm and the PageRank (PR) algorithm are regarded as the benchmark algorithms.

5.4. Evaluation metrics

In the experiments for OCD algorithm, the average false negative (AFN) rate, the average false positive (AFP) rate and the average error cost (AEC) are taken as the metrics to value the performance of the OCD algorithm.

$$AFN = \sum_{i=1}^{Num_t} \left[\left(Num_i^{FN} / Num_p \right) / Num_t \right], \tag{23}$$

$$AFP = \sum_{i=1}^{Num_t} \left[\left(Num_i^{FP} / Num_p \right) / Num_t \right], \tag{24}$$

$$AEC = CAFN \times AFN \times P_t + CAFP \times AFP \times (1 - P_t), \tag{25}$$

where *AFN*, *AFP* and *AEC* denote the average false negative rate, the average false positive rate and the average error cost, respectively. Num_t is the number of topics. Num_i^{FN} is taken as the number of posts that are related to topic *i* but undiscovered. Num_i^{FP} is defined as the number of posts that are not related to topic *i* but regarded to belong to topic *i*. Num_p is the total number of the posts. *CAFN* is the false negative cost, and *CAFP* is the false positive cost. P_t is the new topic generation probability.

In the experiments for OLDUIEA algorithm, the coverage rate (CR) is regarded as the metric to evaluate the performance of OLDUIEA algorithm.

$$CR = \frac{Num_{LR}}{Num_u^i}, \tag{26}$$

where Num_{LR} is taken as the number of top "*n*th percentile" opinion leaders who are related to the topic *i*, and Num_u^i is the number of users who are related to the topic *i*.

5.5. The experimental results

5.5.1. The experimental results for OCD algorithm

Similarity threshold is the important parameter that is used to determine the integrated similarity of two users and identify the opinion communities. In order to achieve the optimal value of the similarity threshold, we discuss the relationship between the metrics and the similarity threshold. Figs. 5–7 describe the value of AEC, AFN and AFP with different similarity thresholds, respectively. Obviously, with the increasement of the similarity threshold, the value of each metric decreases firstly then increases. When the similarity threshold is 0.3, each metric achieves the minimum value. Similarity threshold is

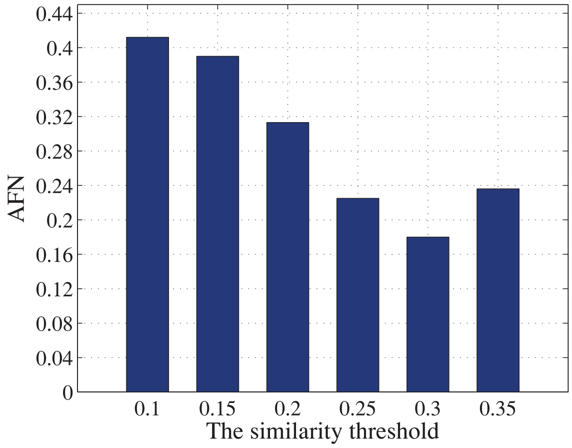

Fig. 6. The value of AFN with different similarity thresholds.

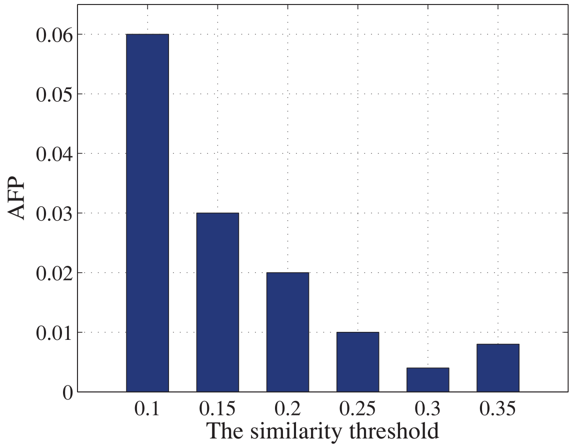

Fig. 7. The value of AFP with different similarity thresholds.

Table 6
The top 5 topics with most posts.

Topic	The number of posts	Keywords
Putin Missing	8650	Coup, Health, Putin, disappearance, House Arrest
Malaysia Airlines crash	6481	Assault, Conspiracy, MH370, Rescue, Malaysia Airlines
Jackie Chan Duang	5256	Shampoo, Jackie Chan, Sliding Shoes, Peng Mei Long, Divine Tune
Spring Festival Gala	4986	Spring Festival Gala, New Year, Go home, Spring Festival, Debunk
Lee Kuan Yew died	3245	Lee Kuan Yew, Die, State Funeral, Hero, Yearn

the key factor to identify the opinion communities. With the increasement of the similarity threshold, the probability that one post or user belongs to one certain topic increases. Therefore, by comprehensively considering the three factors of AEC, AFN and AFP, the similarity threshold should be set to 0.3.

The OCD algorithm is used to cluster 11.713 thousand posts, and 8976 topics are achieved. The top five topics are “Putin Missing”, “Malaysia Airlines crash”, “Jackie Chan Duang”, “Spring Festival Gala” and “Lee Kuan Yew died”, respectively. They have 8650, 6481, 5256, 4986 and 3245 posts, successively. The keywords for each topic also are achieved, which are shown in Table 6. From Table 6, we can know that the posts about “Putin Missing” are most, which implies that there are many users paying attention to the topic of “Putin Missing”. Moreover, they hold a hot discussion about the topic of “Putin Missing”. By comparison, the posts about “Lee Kuan Yew died” are less, i.e., there are fewer users focusing on the topic of “Lee Kuan Yew died”.

Table 7 describes the value of AFN and AFP by varying the number of posts, respectively. With the increasement of the number of posts from 5000–40,000, the value of AFN and AEC increases, respectively. Then, when the number of posts changes from 40,000–50,000, the value of both AFN and AFP ranges between 0.180 and 0.200. Figs. 8 and 9 depicts the value of AFN and AEC with the change of the number of posts, respectively. Obviously, with the increasement of posts, the

Table 7
AFN and AEC with different number of posts.

The number of posts	5000	10,000	15,000	20,000	30,000	35,000	40,000	45,000	50,000
AFN	0.658	0.550	0.472	0.320	0.250	0.200	0.180	0.182	0.185
AEC	0.740	0.630	0.536	0.425	0.352	0.252	0.180	0.200	0.190

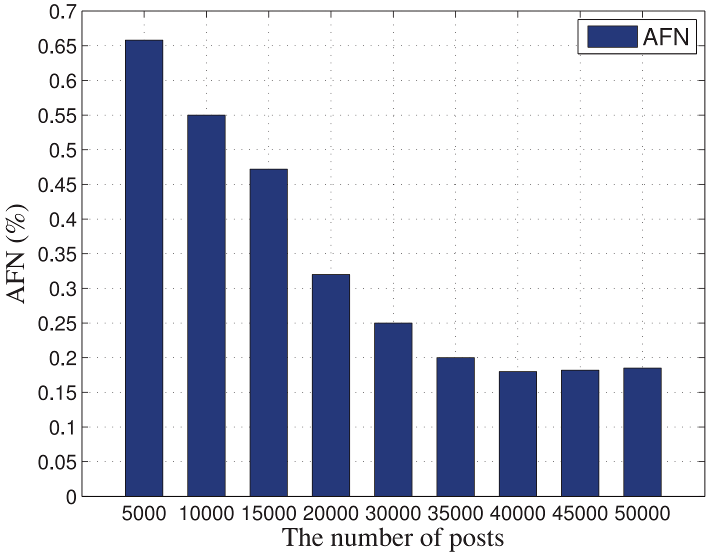

Fig. 8. AFN with different number of posts.

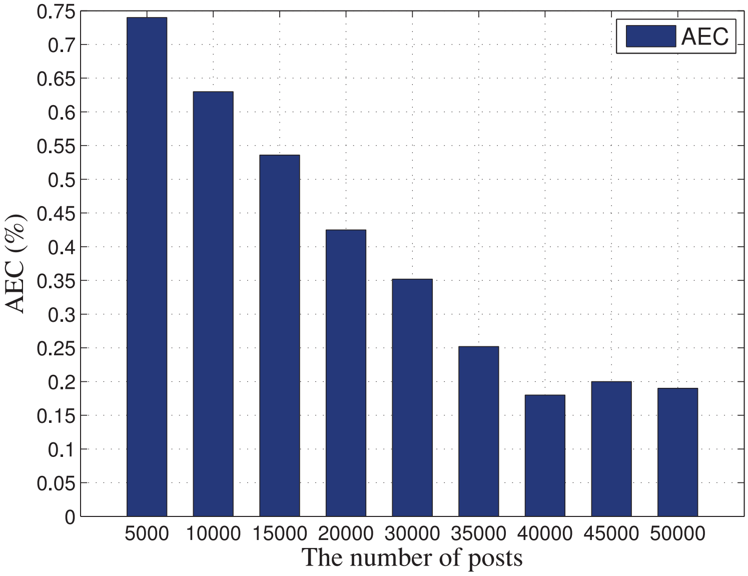

Fig. 9. AEC with different number of posts.

value of AFN and AEC sharply decrease. When the number of posts is more than 40,000, the change of the value of both AFN and AEC tends to stabilization. It implies that the value of AFN and AEC are optimal when the number of posts is more than 40,000. Also, it means that the performance of OCD algorithm becomes better with the increasement of the number of posts, i.e., OCD algorithm is more suitable for the big data process of stream document.

Table 8 describes the value of AFN with different algorithms for the top five topics. 98,650, 93,481, 83,256, 89,735 and 5425 topics are related to “Putin Missing”, “Malaysia Airlines crash”, “Jackie Chan Duang”, “Spring Festival Gala” and “Lee Kuan Yew died”, respectively. The performance of SP algorithm is less than that of OCD algorithm in terms of AFN. Fig. 10 intuitively shows the comparison result of the two algorithms in terms of AFN. The blue rectangle represents the AFN

Table 8
AFN with different algorithms for the top five topics.

Topic	The number of topics	AFN	
		SP algorithm	OCD algorithm
Putin Missing	98,650	0.6534	0.5376
Malaysia Airlines crash	93,481	0.8095	0.8035
Jackie Chan Duang	83,256	0.8143	0.7245
Spring Festival Gala	89,735	0.6935	0.5743
Lee Kuan Yew died	5425	0.7841	0.7897

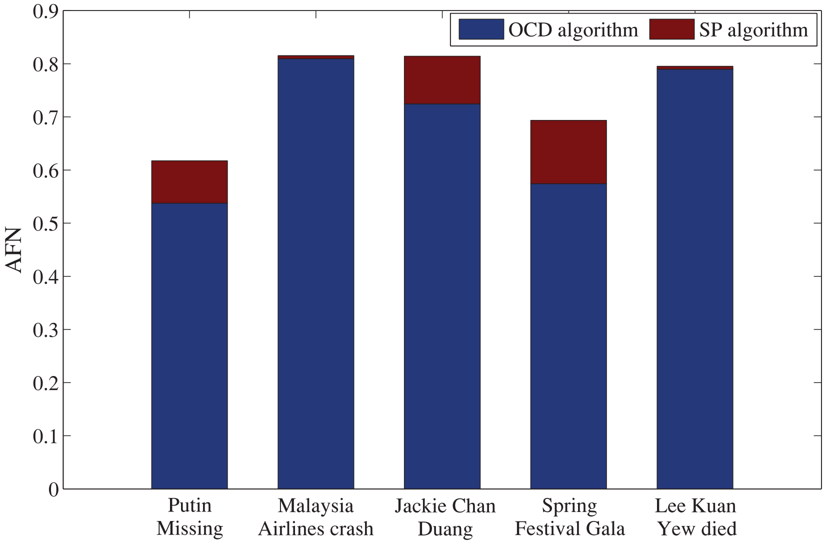

Fig. 10. AFN with different algorithms for the top five topics (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

value of OCD algorithm. The red rectangle above the blue rectangle is taken as the difference between the AFN value of SP algorithm and OCD algorithm. Obviously, the performance of OCD algorithm is better than that of SP algorithm. For example, SP algorithm achieves 12.39% AFN improvement over that of OCD algorithm in terms of “Jackie Chan Duang” topic. Moreover, the AFN value of SP algorithm is improved up to 20.76% comparing with that of OCD algorithm in terms of “Spring Festival Gala”. However, for “Lee Kuan Yew died” topic, the AFN value of SP algorithm achieves 0.56% reduction over that of OCD algorithm. This also implies that OCD algorithm is more suitable for the big data process of stream document. When the number of topics is less, the performance of OCD algorithm cannot be obviously shown.

In order to evaluate the performance of OCD algorithm in a parallel environment (PE), KM algorithm and OCD algorithm, which are run on a single machine environment (SME), are taken as the compared algorithms. Fig. 11 depicts the comparison results of three opinion community detection algorithms in terms with different metrics. The blue rectangle represents the metric value of PE algorithm. The green rectangle above the blue rectangle is taken as the difference between the metric value of SME algorithm and PE algorithm. The red rectangle above the green rectangle is taken as the difference between the metric value of KM algorithm and SME algorithm. Obviously, the performance of KM algorithm is less than that of the other two algorithms. For example, OCD algorithm within SME achieves 39.68% AEC reduction, 40.17% AFN reduction and 61.11% AFP reduction, respectively, comparing with KM algorithm. Moreover, OCD algorithm within PE achieves 47.18% AEC reduction, 45.58% AFN reduction and 77.78% AFP reduction, respectively, comparing with KM algorithm. It implies that KM algorithm has little ability to process the large-scale opinion community clustering problem. OCD algorithm within PE achieves 12.44% AEC reduction, 9.05% AFN reduction and 42.86% AFP reduction, respectively, comparing with OCD algorithm within SME. This implies that OCD algorithm within PE is more suitable for the large-scale opinion community detection.

In this section, the performance of OCD algorithm is evaluated by comparing with that of SP algorithm. Averagely, OCD algorithm can achieve 8.90% AFN reduction comparing with SP algorithm, which means that OCD algorithm is more suitable for the big data process of stream document. Moreover, the performance of OCD algorithm within PE is valued by comparing with that of KM algorithm and OCD algorithm within SME. OCD algorithm within PE can achieve 12.44% AEC reduction, 9.05% AFN reduction and 42.86% AFP reduction, respectively, comparing with OCD algorithm within SME. Moreover, OCD algorithm within PE can achieve 47.18% AEC reduction, 45.58% AFN reduction and 77.78% AFP reduction, respectively, comparing with KM algorithm. This implies that OCD algorithm within PE is more suitable for the large-scale opinion community detection.

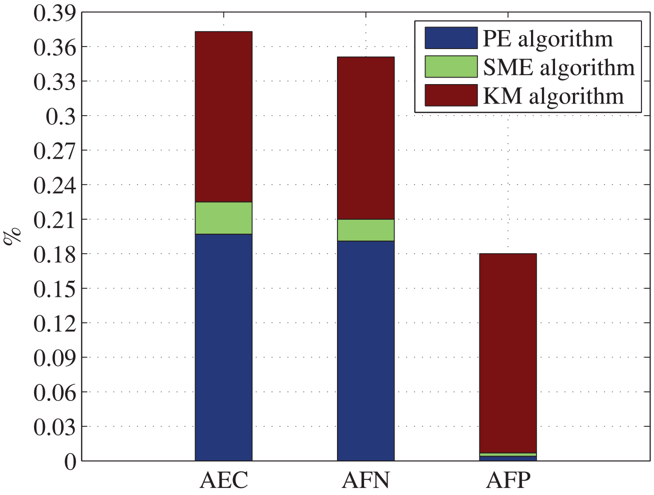

Fig. 11. The comparison results of three opinion community detection algorithms (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

Table 9
The top 10 opinion leaders in different topics.

User Name	Putin Missing			Lee Kuan Yew died		
	User ID	PRV	Percent	User ID	PRV	Percent
User1	89,076	0.42	7.04%	89,076	0.30	5.08%
User2	34,786	0.58	9.72%	34,521	0.85	14.38%
User3	576	0.41	6.87%	908,778	0.46	7.78%
User4	3290	0.37	6.20%	6,740,900	0.70	11.84%
User5	435,886	0.62	10.39%	4356	0.90	15.23%
User6	8,956,432	0.54	9.05%	786,753,291	0.65	11.00%
User7	467,234	0.90	15.08%	467,234	0.50	8.46%
User8	1,245,278	0.78	13.07%	56,321	0.70	11.84%
User9	85,643	0.56	9.38%	0,956,432	0.40	6.77%
User10	96,754	0.79	13.23%	673,328	0.45	7.61%

5.5.2. The experimental results for OLDUIEA algorithm

Table 9 shows the top 10 opinion leaders, which are obtained by OLDUIEA algorithm, in “Putin Missing” topic and “Lee Kuan Yew died” topic, respectively. Obviously, each user has an influence value. The influence value is used to determine the influence of one user. The user with the highest influence value is taken as the opinion leader, because they have more interaction with other users and provide more help for other users. An opinion leader has the power of specialized authority and is at the core of the relationship network of the corresponding topic. Moreover, an opinion leader can control the guidance of public opinion.

Figs. 12 and 13 intuitively describe the top ten users with highest PRV for “Putin Missing” topic and “Lee Kuan Yew died” topic, respectively. From Fig. 12, we can know that the user whose ID number is 467,234, is the opinion leader for “Putin Missing” topic. The activity of the user is more than that of other users. The opinion leader knows more information about “Putin Missing” topic and can provide more help for common users than that of other users. However, the user, whose ID number is 467,234, is not the opinion leader in “Lee Kuan Yew died” topic community. In “Lee Kuan Yew died” topic community, PRV of the user, whose ID number is 467,234, is far less than that of the opinion leader, because the user is unacquainted with “Lee Kuan Yew died” topic or pay less attention to “Lee Kuan Yew died” topic. So, this user has less interaction with other users, and just is a follower. Furthermore, the opinion leader of “Lee Kuan Yew died” topic, i.e. the user 5 in “Lee Kuan Yew died” topic, does not appear in “Putin Missing” topic community. This implies that an opinion leader has certain limitations and cannot have an influence of authority for any topic.

Fig. 14 depicts the CR value with different algorithms. The top thirty users of one certain topic are chosen to evaluate the performance of OLDUIEA algorithm. Obviously, the CR value of OTOLD algorithm and EOLD algorithm is far less than that of OLDUIEA algorithm and PR algorithm. The performance of OLDUIEA algorithm and PR algorithm are similar. This also implies that the network topology structure is very important to the opinion leader detection, because both OLDUIEA algorithm and PR algorithm consider the social network topology structure. For the top 20% opinion leaders, the performance of OLDUIEA algorithm is better than that of PR algorithm. Furthermore, For the top three opinion leaders, the performance of OLDUIEA algorithm is far better than that of PR algorithm. This is because OLDUIEA algorithm considers not only the relationship between users, but the emotional polarity of words in the posts. In addition, OTOLD algorithm and EOLD algorithm just

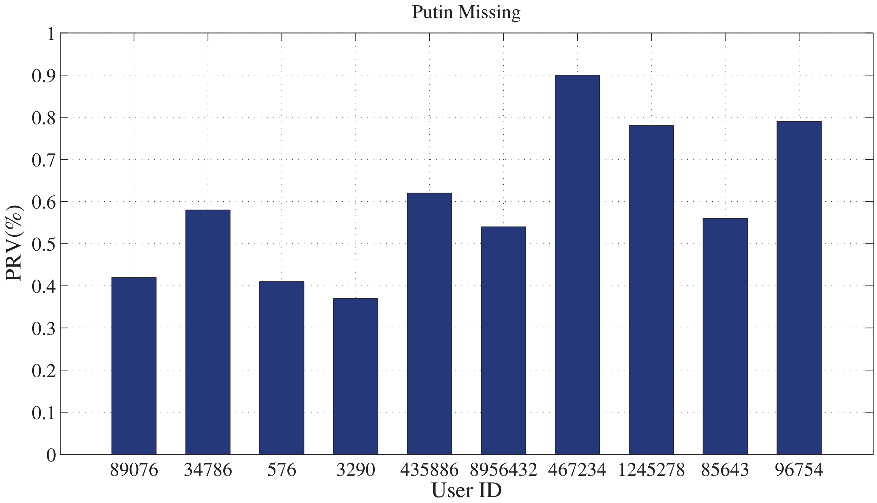

Fig. 12. Top ten users with highest PRV for "Putin Missing" topic.

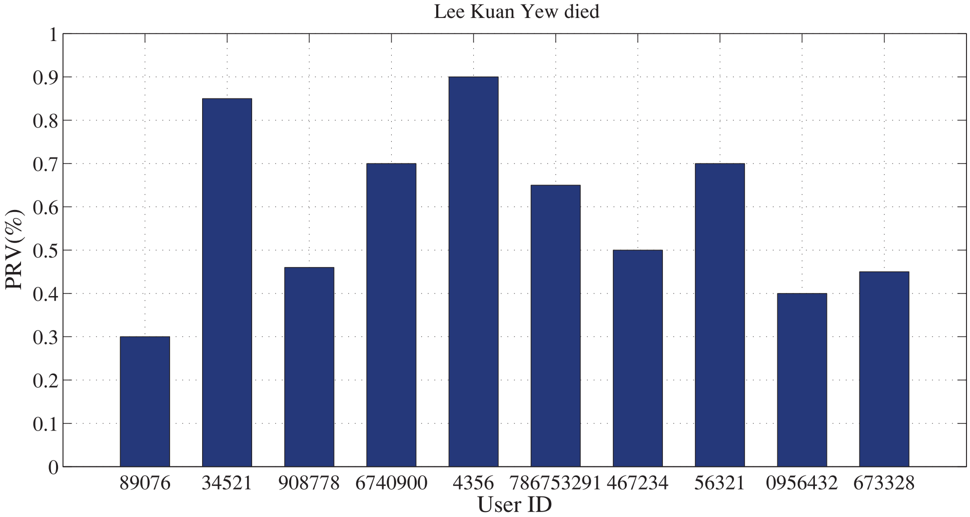

Fig. 13. Top ten users with highest PRV for "Lee Kuan Yew died" topic.

consider the static information of users, such as user ID, user tag, etc., and neglect the relationship between users. So, the CR value of OTOLD algorithm and EOLD algorithm is far less than that of OLDUIEA algorithm.

In this section, in order to illustrate the opinion leader has limited knowledge, the top ten opinion leaders in "Putin Missing" topic and "Lee Kuan Yew died" topic are detected, respectively. The results show that the opinion leader in "Putin Missing" topic is the 5th opinion leader in "Lee Kuan Yew died" topic and has no highest influence on "Lee Kuan Yew died" topic. Meanwhile, the opinion leader in "Lee Kuan Yew died" topic just is a follower in "Putin Missing" topic and doesn't have any feature of the opinion leader. So, an opinion leader has limited knowledge. He/she may be an opinion leader in one topic and be a follower in another topic. Finally, CR is taken as the metric to evaluate the performance of OLDUIEA algorithm. The result shows that the performance of OLDUIEA algorithm is far more than that of OTOLD algorithm and EOLD algorithm. Moreover, for the top 20% opinion leaders, the performance of OLDUIEA algorithm is better than that of PR algorithm. Furthermore, For the top three opinion leaders, the performance of OLDUIEA algorithm is far better than that of PR algorithm.

5.6. Threats to validity

In this section, the relevant threats to the validity of our work are summarized as follows.

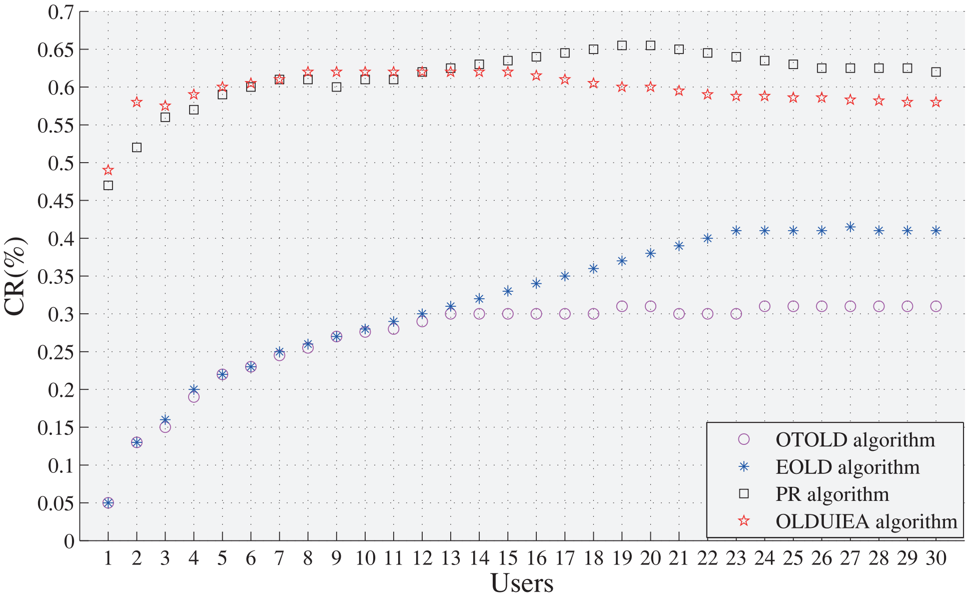

Fig. 14. CR with different algorithms.

5.6.1. Threats to external validity

A first threat to the generalization of our results is related to the size of dataset. In our paper, the cloud platform is applied to improve the computation efficiency of both opinion community detection algorithm and opinion leader detection algorithm. However, if the dataset is small, the advantages of the distributed computing on a cloud platform are not prominent. For this reason, the dataset should be large. In our paper, all data are achieved from the end of the world forum. The web crawler is used to achieve the text information of each user in the forum, such as user ID, the set of posts, the posting time, etc. 11.713 thousand posts are crawled and 8976 topics are achieved. So, the dataset is enough large to present the advantages of the distributed computing environment.

Another consideration is the limited environment of our proposed algorithm. The opinion community detection model and the opinion leader detection model are successively designed based on the forum features. For instance, in the opinion community detection model, the contents of the posts, the posting time and the interaction between users are considered. Especially, by considering the posting time, the time similarity model is built to avoid the topic drift. So, our proposed algorithms are suitable for the opinion community detection and opinion leader detection in Forum.

5.6.2. Threats to internal validity

In this paper, the opinion communities are detected based on the integrated similarity model. The integrated similarity model is built based on the content similarity model, the time similarity model and the interaction relationship model. So, the similarity threshold becomes very important to the performance of the proposed opinion community detection algorithm. This similarity threshold directly affects the experimental results of the proposed opinion community detection algorithm. In order to achieve the optimal value of the similarity threshold, the similarity threshold is discussed and determined by experiments. The experimental discussion and analysis are shown on Paragraph one in Section 5.5.1. Finally, the similarity threshold is set to 0.3.

6. Conclusion

In this paper, an opinion community detection method is proposed by considering the content similarity, the time similarity and the topology structure of users. The vector space model is introduced to mine the keywords of the text information of posts. Also, the keyword weight computation model is applied to calculate the keyword weights of the text information of posts. Then, the similarity models in terms of user contents, time information and topology structure of users are introduced in detail. Finally, the integrated similarity between two users is achieved based on the content similarity, the time similarity and the topology structure of users. Furthermore, the opinion communities are detected based on the integrated similarities.

In order to identify the opinion leader, a method for opinion leader detection is proposed. A directed graph is created to formulate the interaction relationship between users. Then, the user influence model is built based on the content similarity and network topology. The emotional analysis model is presented to determine the emotional tendency of posts, in which the contents of one post are divided into several sentences. Moreover, the emotional tendency of one sentence is determined by the emotional tendencies of the words. Furthermore, the occurrence frequencies of the negative words are also considered. Finally, a model of user influence value is built based on the improved PageRank algorithm. The user with the highest influence value is considered as the opinion leader.

Finally, the performance of the proposed algorithms is evaluated in a distributed computing environment. Meanwhile, the extensive experiments are conducted. The results indicate that our proposed opinion community detection algorithm can effectively detect the opinion communities. Also, the proposed opinion leader detection algorithm can significantly identify the opinion leader in social networks. In the future works, the opinion community detection algorithm and the opinion leader detection algorithm, which are suitable for the dataset on large-scale and the generalized social network platforms, will be studied.

Conflict of interest

None.

Acknowledgements

This work was supported by the National Natural Science Foundation (NSF) under grants (No. 61672397, No. 61871352), Application Foundation Frontier Project of WuHan (No. 2018010401011290), The Fundamental Research Funds for the Central Universities (WUT No. 2019-YB-028), National Key R&D Program of China (No. 2018YFE0101000) and Open Fund of Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University (Grant No. 2019KEY003). Beijing Intelligent Logistics System Collaborative Innovation Center Open Project (No. BILSCIC-2019KF-02), Beijing Youth Top-notch Talent Plan of High-Creation Plan (No. 2017000026833ZK25), Canal Plan-Leading Talent Project of Beijing Tongzhou District (No. YHLB2017038), Grass-roots Academic Team Building Project of Beijing Wuzi University No. 2019XJJCTD04. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

References

[1] W. Ai, D.P. Li, Parallelizing hot topic detection of microblog on spark, in: Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD, Changsha, China, 2016, pp. 1461–1468. August 13–15.
[2] K.N. Aini, I. Najahaty, L. Hidayati, H. Murfi, S. Nurrohmah, Combination of singular value decomposition and k-means clustering methods for topic detection on twitter, in: Proceedings of the International Conference on Advanced Computer Science and Information Systems, ICACSIS, Depok, Indonesia, 2015, pp. 123–128. October 10–11.
[3] Y. Chen, X.L. Wang, B.Z. Tang, R.F. Xu, B. Yuan, X. Xiang, J.Z. Bu, Identifying opinion leaders from online comments, in: Proceedings of the Chinese National Conference on Social Media Processing, SMP, Beijing, China, 2014, pp. 231–239. November 1–2.
[4] Y.C. Ho, H.M. Liu, H.H. Hsu, C.H. Lin, Y.H. Ho, L.J. Chen, Automatic opinion leader recognition in group discussions, in: Proceedings of the Conference on Technologies and Applications of Artificial Intelligence, TAAI, Hsinchu, Taiwan, 2016, pp. 138–145. November 25–27.
[5] L.C. Jiang, B. Ge, W.D. Xiao, M.Z. Gao, BBS opinion leader mining based on an improved pagerank algorithm using mapreduce, in: Proceedings of the Chinese Automation Congress, Changsha, China, 2013, pp. 392–396. November 7–8.
[6] C. Juan, C. Angel, G.S. Ana, A step forward for topic detection in twitter: an FCA-based approach, Expert Syst. Appl. 57 (2016) 21–36.
[7] G. Lee, U. Yun, Single-pass based efficient erasable pattern mining using list data structure on dynamic incremental databases, Future Gener. Comp. Syst. 80 (2017) 12–28.
[8] C.L. Li, J.P. Bai, J.H. Tang, Joint optimization of data placement and scheduling for improving user experience in edge computing, J. Parallel Distr. Comput. 125 (2019) 93–105.
[9] C.L. Li, J.P. Bai, W.J. Zhao, X.H. Yang, Community detection using hierarchical clustering based on edge-weighted similarity in cloud environment, Inform. Process. Manag. 56 (1) (2019) 91–109.
[10] C.L. Li, J.H. Tang, H.L. Tang, Collaborative cache allocation and task scheduling for data-intensive applications in edge computing, Future Gener. Comput. Syst. 95 (2019) 249–264.
[11] C.L. Li, J. Zhang, T. Ma, H.L. Tang, L. Zhang, Y.L. Luo, Data locality optimization based on data migration and hotspots prediction in Geo-distributed cloud environment, Knowl. Based Syst. 165 (2019) 321–334.
[12] H. Li, Q. Li, Forum topic detection based on hierarchical clustering, in: Proceedings of the International Conference on Audio, Language and Image Processing, ICALIP, Shanghai, China, 2016, pp. 529–533. July 11–12.
[13] H. Lin, B. Sun, J.J. Wu, H.T. Xiong, Topic detection from short text: a term-based consensus clustering method, in: Proceedings of the 13th International Conference on Service Systems and Service Management, ICSSSM, Kunming, China, 2016, pp. 1–6. June 24–26.
[14] M. Pavkovic, J. Protic, Intelligent crawler for web forums based on improved regular expressions, in: Proceedings of the 21st Telecommunications Forum Telfor, TELFOR, Belgrade, Serbia, 2013, pp. 817–820. November 26–28.
[15] Y.H. Rao, J.H. Pang, H.R. Xie, L. An, T.L. Wong, Q. Li, F.L. Wang, Supervised intensive topic models for emotion detection over short text, in: Proceedings of the International Conference on Database Systems for Advanced Applications, DASFAA, Suzhou, China, 2017, pp. 408–422. May 21–24.
[16] S. Rill, D. Reinel, J. Scheidt, R.V. Zicari, Politwi: early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis, Knowl. Based Syst. 69 (2014) 24–33.
[17] S. Shahrivari, S. Jalili, Single-pass and linear-time k-means clustering based on mapreduce, Inf. Syst. 60 (2016) 1–12.
[18] C. Wang, Y.J. Du, M.W. Tang, Opinion leader mining algorithm in microblog platform based on topic similarity, in: Proceedings of the 2nd IEEE International Conference on Computer and Communications, ICCC, Chengdu, China, 2016, pp. 160–166. October 14–17.
[19] G.H. Wang, Y.J. Liu, J.M. Li, X.Y. Tang, H.B. Wang, Super edge coupling algorithm and its application in coupling mechanism analysis of online public opinion super network, Expert Syst. Appl. 42 (5) (2015) 2808–2823.
[20] J.F. Wang, X. Jia, L.B. Zhang, Identifying and evaluating the internet opinion leader community through K-clique clustering, J. Comput. 8 (9) (2014) 2284–2289.
[21] M.S. Wang, P.P. Jayaraman, E. Solaiman, L.Y. Chen, Z. Li, S. Jun, D. Georgakopoulos, R. Ranjan, A multi-layered performance analysis for cloud-based topic detection and tracking in big data applications, Future Gen. Comput. Syst. 87 (2018) 580–590.
[22] W. Xie, F.D. Zhu, J. Jiang, E.P. Lim, K. Wang, Topicsketch: real-time bursty topic detection from twitter, IEEE Trans. Knowl. Data Eng. 28 (2016) 2216–2229.
[23] M. Xing, Tianya Forum, 2018, http://bbs.tianya.cn/.

[24] D. Yan, E. Hua, B. Hu, An improved single-pass algorithm for chinese microblog topic detection and tracking, in: Proceedings of the IEEE International Congress on Big Data, BigData Congress, San Francisco, CA, USA, 2016, pp. 251–258. June 27, - July 2.
[25] H. Ye, J.P. Du, Opinion leader mining of social network combined with hierarchical sentiment analysis, in: Proceedings of the Chinese Intelligent Automation Conference, CIAC, Tianjin, China, 2017, pp. 639–646. June 2–4.
[26] C. Zhang, H. Wang, L. Cao, W. Wang, F. Xu, A hybrid term-term relations analysis approach for topic detection, Knowl. Based Syst. 93 (2016) 109–120.
[27] E. Zhou, N. Zhong, Y.F. Li, Extracting news blog hot topics based on the W2T methodology, World Wide Web 17 (3) (2014) 377–404.
[28] X.J. Zhou, X.J. Wan, J.G. Xiao, CMINER: opinion extraction and summarization for chinese microblogs, IEEE Trans. Knowl. Data Eng. 28 (7) (2016) 1650–1663.