Detecting Opinion Leader Dynamically in Chinese News Comments*

Kaisong Song[1], Daling Wang[1,2], Shi Feng[1,2], and Ge Yu[1,2]

[1] School of Information Science and Engineering, Northeastern University
[2] Key Laboratory of Medical Image Computing (Northeastern University), Ministry of Education, Shenyang 110819, P.R. China
songkaisongabc@126.com,
{wangdaling,fengshi,yuge}@ise.neu.edu.cn

Abstract. Nowadays, more and more users publish their opinions by means of Web 2.0. Analyzing users' opinion, discovering the relationship between opinions, and detecting opinion leader are important for Web public opinion analysis. Opinion leader is regarded as the most influential comment and user during the information dissemination process. However, most existing researches pay less attention on their internal relation, implicit relationship between users' opinions and the changes of opinion leader over time. In this paper, we focus on modeling comment network with explicit and implicit links for detecting the most influential comment dynamically, and modeling user network and clustering users for detecting the most influential user. We propose an approach with sentiment analysis, explicit and implicit link mining for modeling comment network in Chinese news comments. We also propose an algorithm for detecting most influential comment from the comment network dynamically. Moreover, we model user network based on the comments network, and detect the most influential user from the user network. Experiments using Chinese Sina news comment dataset show that our approach can detect opinion leaders and the changes of them over time dynamically.

Keywords: Opinion mining, opinion leader, comment network, sentiment analysis, clustering.

1 Introduction

In recent years, with the advent of Web 2.0, more and more users publish their opinions about social events and phenomena by means of tools such as blogs, microblogs, and forums, among which news comments contain rich public opinions. In these opinions, opinion leaders often provide constructive information for other users, and play a critical role during the information dissemination process. Analyzing

* Project supported by the State Key Development Program for Basic Research of China (Grant No. 2011CB302200-G) and National Natural Science Foundation of China (Grant No. 60973019, 61100026).

L. Wang et al. (Eds.): WAIM 2011 Workshops, LNCS 7142, pp. 197–209, 2012.

users' opinions, discovering the relationship between opinions, especially detecting opinion leaders are important for Web public opinion analysis. The researchers have done some pioneering work for discovering opinion leaders in Web 2.0 media. However, there are still some drawbacks in these researches:

(1) Most existing work pay more attentions on analyzing explicit relationship among users' opinions such as following, replying, but neglecting the implicit ones.

(2) A user's' opinion may be a gradual process and affected to a large extent by previous users. However, most existing methods for detecting opinion leaders have not considered the impact of both time and sentiment.

(3) Few people consider the internal relation between the comment's influence and user's one for detecting the most influential user.

(4) In addition, opinion leader with positive influence is more useful in real life.

Based on the characteristics of news comments, in this paper, we focus on modeling comment network with explicit and implicit links for detecting the most influential comment from the comment network dynamically, and clustering users for the most influential user. We model comment network by sentiment orientation analysis, opinion similarity calculating, explicit and implicit link mining, and other related techniques with opinion mining, and propose an algorithm named Dynamic OpinionRank for detecting the most influential opinion from the comment network. Moreover, we apply DBSCAN [3] to find the most influential user. According to the method proposed above, we detect the opinion leader including user and comment. The experiments using Chinese Sina news comments dataset validate the effectiveness of the proposed dynamical opinion leader detection algorithm.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 describes the problem definition. Section 4 analyzes sentiment orientation and model the comment network. Section 5 detects opinion leader including the most influential comment and user. Section 6 shows our experiment results. Finally Section 7 concludes the research and gives directions for future studies.

2 Related Work

In this paper, our purpose is to detect opinion leader in Web 2.0 social media. In this field, the researchers have done some pioneering work. Xiao et al. [9] propose a LeaderRank algorithm to identify the opinion leaders in BBS, which includes finding the interest user group based on topic content analysis and defining the authority value as the weight of the link between users. They also utilize LeaderRank algorithm to identify opinion leaders based on community discovery and emotion mining methods [10]. Freimut and Carolin [5, 6] present an approach, which initially detects opinions and relationships among forum users, extracts main influential factors for opinion forming in virtual communities, and identifies opinion leaders and analyzes opinion evolvement by social network analysis. Zhou et al. [12] introduce the concept of Opinion Networks and propose OpinionRank algorithm to rank the nodes in an opinion network. Zhai et al. [11] propose interest-field based algorithms to identify opinion leaders in BBS, which not only take into account of the reply networks' structure but also the users' interest space. Feng and Timon [4] proposes a framework, which builds ontology for a marketing product to identify opinion leaders using the information retrieved from blog content, authors, readers, and their relationships.

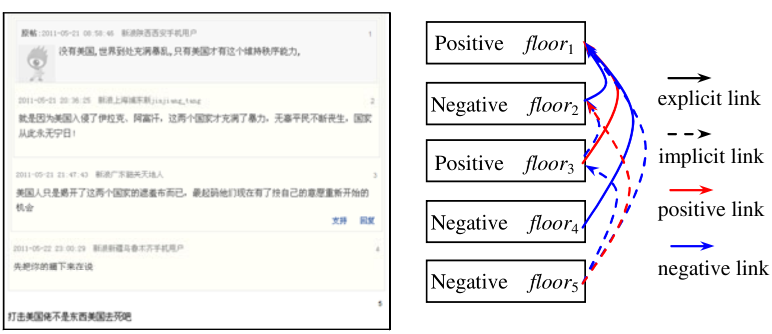
These studies above have some obvious shortcomings. Firstly, few of them do research based on Chinese. Secondly, although some of the works take emotional factors into consideration, they do not discover the impact of other features such as time. To overcome these shortcomings, in this paper we propose a dynamic detection technology for finding opinion leaders from Chinese news comments.

3 Problem Description

Let $C=\{C_1, C_2, \ldots, C_n\}$ be a comment set, and C_i ($1 \le i \le n$) be an item of comment. We can obtain the sentiment orientation O_i ($1 \le i \le n$) for every $C_i \in C$ by sentiment analysis techniques, and the value of O_i is defined as P, N, and M corresponding to positive (support), negative (oppose), and neutral sentiment, respectively. Moreover, we give the following definitions.

Definition 1 (explicit link and implicit link). For C_i and C_j ($1 \le i,j \le n$), suppose C_i be published earlier than C_j. If C_j is a follower or reply of C_i, C_j is regarded as having an explicit link to C_i. If they don't have the relationship, but C_i has semantic similarity with C_j (same or different), C_j is regarded as having an implicit link to C_i.

Definition 2 (positive link and negative link). If C_j has the same sentiment orientation with C_i, the link (explicit or implicit) is called as "positive link", otherwise as "negative link". According to this relationship among comments, we can obtain a link structure about C. Fig.1. shows a comment webpage and its link structure in the webpage.



(a) Comment page (b) Link Structure

Fig. 1. An Example of a Comment Page and Its Linked Structure

In Fig.1, $floor_1$ is the first reviewer, $floor_2$, $floor_3$ and $floor_4$ are the followers of $floor_1$, and $floor_5$ represents user himself. So $floor_2$, $floor_3$ and $floor_4$ are explicitly linked to $floor_1$. According to their content, $floor_5$ is linked implicitly to others.

Definition 3 (comment network and user network). Let E be the set of the links of definition 1 and definition 2 (as edge), and V be the comment set, i.e. $C=\{C_1, C_2, \ldots, C_n\}$ (as vertex), then G_{CN} (V, E) be a comment network (as graph). Moreover, if the vertex in G_{CN} (V, E) is defined with the user who published comments instead of comment, G_{CN} (V, E) will express the relationship among users. We named it as $G_{UN}(V, E)$, i.e. user network. Different form G_{CN} (V, E), the edges in $G_{UN}(V, E)$ are only explicit links.

$G_{CN}(V, E)$ and $G_{UN}(V, E)$ are directed graphs similar to link structure in Fig.1 (b). Difference from Fig.1 (b), the edge in $G_{CN}(V, E)$ or $G_{UN}(V, E)$ has a weight wt, and wt can be calculated according to similarity of two vertexes linking the edge.

Our final purpose is to discover opinion leader including the most influential comment in $G_{CN}(V, E)$ and the most influential user in $G_{UN}(V, E)$, and they may change over time. So we call the process as "detecting opinion leader dynamically".

For this purpose, we will do the following work.

(1) Analyzing sentiment orientation of every comment.
(2) Constructing the link structure as Fig.1 (b).
(3) Modeling $G_{CN}(V, E)$ and $G_{UN}(V, E)$ based on links between comments and time.
(4) Detecting opinion leaders including the most influential comment and user.

Above work will be introduced in Section 4, Section 5 and Section 6 in this paper.

4 Sentiment Analysis and Comment Network Modeling

After downloading related comments about news, we can find explicit links among the comments according to webpage structure. Then we will analyze the content of every comment and calculate the similarity between comments for analyzing their sentiment orientation, finding implicit links, determining positive links and negative links, obtaining the weight of every edge, and finally building the comment network.

4.1 Sentiment Analysis

We utilize HowNet [2] sentiment lexicon for judging sentiment orientation.

Firstly, every comment C_i $(1 \le i \le n)$ is partitioned into m sentences i.e. $<S_1, S_2, S_3, \ldots, S_m>$, and apply ICTCLAS [7] to splitting sentence into l words i.e. $<w_1, w_2, w_3 \ldots w_l>$. After that, we extract adjectives, nouns, verbs with sentiment and match them according to the lexicon. Thirdly, we calculate the total number of negation words such as "no", "not". If the number of negation words is odd, the emotional orientation will remain unchanged. If it is even, the sentiment orientation will switch to the opposite. Finally, we get the value of sentiment orientation 1, -1 and 0 according to the rules shown below. We describe the algorithm as Algorithm CommentOrientation.

Algorithm CommentOrientation;
Input: C is any C_i;
Output: O as the sentiment orientation of C and $O \in \{P, N, M\}$;
//$\{P, N, M\}$ corresponds to {positive, negative, neutral}
Description:

1. partition C into $<S_1, S_2, S_3, \ldots, S_m>$; //$S_i$ ($1 \le i \le m$) is a sentence
2. for each sentence S_i in C;
3. {splitting S_i into $<w_1, w_2, w_3 \ldots w_l>$; //$w_i$ ($1 \le i \le l$) is a word
4. for every sentiment word $w_{ij} \in <w_{i1}, w_{i2}, w_{i3} \ldots w_{il}>$
5. w_{ij}.value={1, -1, 0}| w_{ij}.orientation=$\{P, N, M\}$;
6. compute sentiment orientation of S_i by $S_i.orientation = \sum_j w_{ij}.value$
7. for negation word set $N=<n_{i1}, n_{i2}, n_{i3} \ldots>$ ($n_{ik} \in <w_{i1}, w_{i2}, w_{i3} \ldots w_{il}>$)
8. if $|N|$ is an odd then S_i.orientation=-1* S_i.orientation;
9. }
10. compute orientation C by $C.orientation = \sum_{i=1}^{m} S_i$;
11. $O=\{P, N, M\}|C$.orientation={>0, <0, =0};

4.2 Implicit Link Discovery

Through webpage structure, we can find explicit links easily. Compared with explicit links detection, implicit relationships are much more difficult to find out. According to definition 1, we have to analyze content of every comment for obtaining implicit links. Here we use Vector Space Model (VSM) to describe every comment and calculate the similarity between comments to get their implicit links.

In detail, for two comments C_i and C_j, we remove punctuation and eliminate stopwords from C_i and C_j. Then we split C_i and C_j into words respectively, and apply a vector consisting of meaningful characteristic words to describe C_i and C_j. After that, we utilize $wt_{ij}=TF_{ij} \times IDF_i$ to calculate weighting by document frequency and inverse document frequency. Here TF_{ij} can be normalized by using $0.5+0.5 \times (TF_i/MaxTF_i)$. In the end, two formulas $1+\log(TF)$ and $1+\log(N/DF)$ can be utilized for smoothing the results. After that, we set a threshold for judging whether C_i and C_j have implicit link. And if the threshold is satisfied, we say there is an implicit link $C_i \rightarrow C_j$ (if C_j is published earlier than C_i) or $C_j \rightarrow C_i$ (if C_i is published earlier than C_j).

After above process, we can generate all explicit links and implicit links between any C_i and C_j in comment set C.

4.3 Comment Network Modeling

Based on the results of sentiment analysis, explicit and implicit links mining, we can give out positive links and negative links for comment set C according to definition 2 and further build comment network according to definition 3. The process is described as Algorithm CommentNetworkBuild.

Algorithm CommentNetworkBuild;
Input: explicit links and implicit links in C, sentiment orientation O_i of every $C_i \in C$;
Output: $G_{CN}(V, E)$ //Comment Network of C;
Description:

1. for each $C_i \in C$
2. for each $C_j \neq C_i \in C$
3. if (C_i link to C_j) //the link includes explicit and implicit link
4. if C_i has the same sentiment orientation with C_j
5. C_i positive link to C_j;
6. else
7. C_i negative link to C_j;
8. assign weight wt_{ij} for edge $C_i \rightarrow C_j$;

In the Algorithm, Line 8 is for assigning weight for every edge with positive links and negative links. Initially, wt_{ij} is calculated with formula (1).

$$wt_{i,j}=tag \times similarity(C_i,\ C_j)\ \ (-1 \leq wt_{i,j} \leq 1) \tag{1}$$

The function $similarity(C_i, C_j)$ represents the similarity between comment C_i and C_j, and tag means the relationships of them. If $C_i \rightarrow C_j$ is positive link, tag=1. If $C_i \rightarrow C_j$ is negative link, tag=-1.

In fact, wt_{ij} can also express influence of C_j to C_i. Each comment is accompanied by the attribute of time, i.e. time of the comment published, and the relationship between two comments is related to time. For example, when we are reading comments, we may prefer to read those comments that published recently rather than the old ones. So we can come to a conclusion that the impact of time should be taken into consideration. As is said above, we propose a model, which is shown in Fig.2 to further explain the idea.

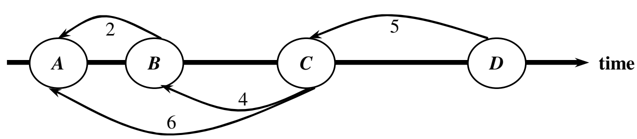


Fig. 2. Time Interval among Comments

As is shown in Fig.2, the former published comments can influence the latter ones. For instance, $B \rightarrow A$ means B is influenced by A. The distance between two comments is their time interval. The larger the distance is, the less possibility the latter will be influenced. For example, the time distance between C and A is larger than the distance between C and B, so we think that C prefers to be influenced by B rather than A.

For comments, the opinion leader is changing over time, so wt_{ij} will concern similarity of C_i with C_j and the time interval between them.

5 Opinion Leader Detection

Opinion leader can be regarded as the most influential comment or its publisher, i.e. user. In this paper, we can detect the most influential comment from $G_{CN}(V, E)$ by computing the score rank of comments, and further build user network $G_{UN}(V, E)$ and discover the most influential user from $G_{UN}(V, E)$.

5.1 The Most Influential Comment Detection

In the field of Web search and Web link analysis, PageRank algorithm [1] is used widely and has excellent effects in many applications. Inspired by PageRank algorithm, we propose a new random walk model called Dynamic OpinionRank, which take emotional and temporal features into consideration. In the meanwhile, a concept of opinion similarity has been proposed and the linked relationships among comments are completely different from the original page linked relationships.

News comments are usually displayed in the form of paging technique. When a user publishes comment, he may be influenced by other comments on two kinds of browsing patterns. Firstly, if the user browses comments backwards and forwards randomly, he will be affected by random comments. Secondly, if he is interested in a certain topic, he will focus on it and may search for relevant comments. That is to say, a user may be influenced by a fixed topic and expresses positive or negative opinions towards relevant comments. So $G_{CN}(V, E)$ may transfer at a certain probability.

Considering users' general browsing habits, our algorithm is based on a reasonable hypothesis that users like referring to the newly published opinions. If the released time of a comment is far from now, the user's comment has less probability to be influenced. Similarly, comments may have two probabilities below to be influenced:

(1) They will be influenced by a relevant topic in a probability f to some purpose.
(2) They may have a probability 1-f to be influenced randomly.

The definition of f is shown as formula (2).

$$f(t_1, t_2, D) = D^{\frac{|t_2 - t_1| \times K}{60 \times 1000 \times 60}} \tag{2}$$

According to the formula above, f is a function relevant to damping coefficient D, t_1 and t_2 represent published time (with the unit of microseconds) of former and latter comment respectively, so f is an alterable damping coefficient. If the cited comment is far from now, it has less probability to be visited. K is a control parameter and we initialize it to 2. D is a damping coefficient which is unrelated to time and we initialize it to 0.85. When a comment is influenced explicitly, it will be less influenced as $|t_2\text{-}t_1|$ becomes bigger. Otherwise it is influenced more deeply. If $|t_2\text{-}t_1|<1/K$, f is bigger than D. If $|t_2\text{-}t_1|>1/K$, f is smaller than D. And if there is no relationship, f is D. Based on the above theory, a promoted model which is similar to PageRank has been proposed as formula (3).

$$P = \left[(1 - f(t_1, t_2, D)) \frac{E}{n} + f(t_1, t_2, D) A^T \right] P \tag{3}$$

where A is an $n \times n$ matrix which is shown in formula (4) and n represents the total number of comments. The matrix reflects the mutual influences among comments, and a_{ij} means the comment i is influenced by the comment j.

$$A^T = \begin{bmatrix} a_{11} & \cdot & \cdot & a_{n1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{1n} & \cdot & \cdot & a_{nn} \end{bmatrix} \tag{4}$$

As for $a_{ij}=wt_{ij}/OD_i$, OD_i represents out-degree of comment i (i.e. C_i). As comments do not always have a large number of out-degrees and in-degrees in comparison with Web page link analysis, matrix A may be not a transfer matrix. If a node only has in-degree but has not out-degree, OD_i will be 0. We adopt general processing method similar to PageRank, but double the result after taking sentiment features into consideration. And we use the formula $a_{ij}=(1/n)^2$ to initialize the line of a_{ij}.

We utilize the formula (5) for calculating final scores of C_i.

$$P(i) = \sum_{1 \le k \le n} \frac{(1 - f(t_i, t_k, D))}{n} + \sum_{(j,i) \in E} f(t_i, t_j, D) \frac{P(j) \times wt_{ji}}{OD_j} \tag{5}$$

where $P(i)$ presents authority value. If we follow the above methods, we will get the final ranking score of each comment in a period of time. Afterwards, we will select the top comment with the highest score as the opinion leader. According to the above operation, a comment network diagram can be modeled easily. Only if a vertex gets more in-degree, which means the comment expresses consistent emotional orientation with others, and time interval is not too long, the comment may get a higher ranking score. The Dynamic OpinionRank algorithm is given below.

Algorithm Dynamic OpinionRank;
Input: $G_{CN}(V, E)$; parameter D and K; threshold ε;
Output: PR //$PR[i]$ is ranking score of comment C_i;
Description:

compute matrix A from $G_{CN}(V, E)$;
Repeat
  for i=1 to n
    {for all j satisfying $a_{ij}\neq 0$ compute $f(t_i, t_j, D)$ using formula (2);
     compute $P(i)$ using formula (5);
    }
Until |current $P(i)$-last $P(i)$|$\leq\varepsilon$;

5.2 The Most Influential User Detection

According to the explicit relationships among comments, we can build up $G_{UN}(V, E)$ to express the relationship among users. If a user replies to another user, then there will be a link between the two users. In $G_{UN}(V, E)$, we can calculate Degree Centrality and Proximity Prestige [8], and get Comment Quality from the ranking score calculated by Dynamic OpinionRank algorithm proposed above. They will be used to detect the influence of a user.

In [8], Degree Centrality and Proximity Prestige are calculated by formula (6) and formula (7) respectively, and reflect the out-degree and range of user i.

$$DC(i) = OD_i/(n-1) \tag{6}$$

$$PP(i) = (|I_i|/(n-1)\Big/(\sum_{i\in I_i} d(j,i)/|I_i|) \tag{7}$$

where I_i is the set linked to vertex (user) i, OD_i is the out-degree of user i (the same definition with comment), and $d(j, i)$ is the shortest link length between i and j. formula (8) is used for getting Comment Quality, and shows influence of comment.

$$CQ(i) = \frac{\sum_{j\in UC_i} Score(j)}{|UC_i|} \times \frac{CL(i)}{CML} \tag{8}$$

As for formula (8), UC_i represents the set of comments published by user i, $Score(j)$ is ranking score of comment j (from Dynamic OpinionRank), $CL(i)$ is the mean value of all comment lengths of user i, and CML is the max length of all users' comments.

Each user is described by $PP(i)$, $CQ(i)$, and $DC(i)$, and the vector represents a point in the three dimensional space. The method below is based on the hypothesis that if a user has strong influence, it will be away from the other points. So we use DBSCAN [3] algorithm to cluster these points, and the outliers may be the most influential users. We set reasonable radius and MinPts for clustering, and then use the formula (9) below to detect the most influential user from the outliers.

$$S(i) = (w_{PP} \times PP(i) + w_{CQ} \times CQ(i) + w_{DC} \times DC(i))/3 \tag{9}$$

where w_{PP}, w_{CQ}, and w_{DC} represent the weight of $PP(i)$, $CQ(i)$, and $DC(i)$, respectively.

6 Experiment Results

In order to get a real dataset, we collect comments from four different period of time. The dataset is about "Libya's civil war". Based on the purpose, we collect real comments about this title from Sina news forum (http://comment4.news.sina.com.cn/ comment/skin/ default.html?channel=gj&newsid=1-1-23003128&style=0#page=10) during two days (2011-08-17 08:04:37~2011-08-18 06:47:08).

6.1 Opinion Leader Detection from Comments

By building $G_{CN}(V, E)$, we apply the most influential comment detection method in Section 5.1, the ranking scores of comments are shown in Fig.3 and Fig.4.

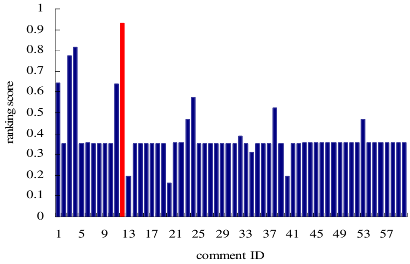


Fig. 3. Ranking (08:04:07-11:07:11)

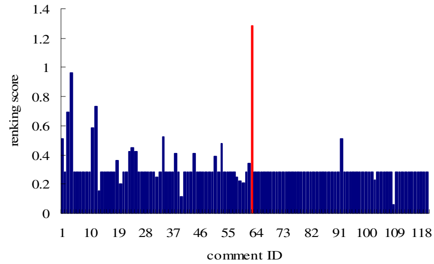


Fig. 4. Ranking (08:04:07-13:45:49)

In Fig.3, a total number of 60 comments are in this period of time. According to the diagram, it is easy to find that NO.12 gets the highest score. However, as there is much less mutual influence between the comments, the opinion leader may change in all probability as time goes on.

From the Fig.4, we can find that NO.12 is not the opinion leader, and NO.63 gets the highest scores, so it is the opinion leader now. Some comments get low value because they are opposed by the other users. Though, there are more comments, mutual influence is still not too much.

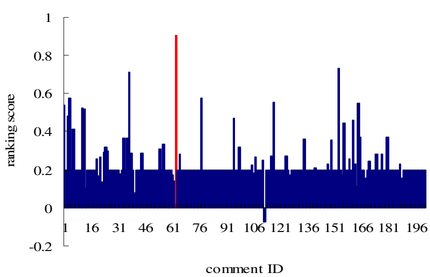


Fig. 5. Ranking (08:04:07-18:43:22)

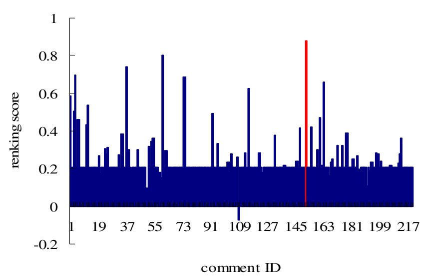


Fig. 6. Ranking (08:04:07-next day 06:47:08)

In the third period of time of Fig.5, NO.63 continues to be the opinion leader, and his scores are decreasing as the time goes on. With the increasing number of comments, newly published comment is increasing much faster, and there is much more mutual influence between comments. This figure shows new comments attract more attention, and it also proves the effectiveness of taking time into consideration.

In the Fig.6, after tracking a day of the news forum, the number of comments reaches to 220. Now NO.151 gets the highest score, and it is the opinion leader. Compared with the Fig.5, many comments' scores are increasing to a certain degree. The number of comments keeps stability in a certain range because of its real-time characteristic, and No.151 is probably the opinion leader in the end.

The most influential comment and ranking score are shown in Table 1 below.

Table 1. The Changes of Opinion Leader in the Comments

No	Content of Comment	Ranking Score
No.12	政府军打击不准确,无非是不伤害平民百姓的	0.9312
No.63	和当年萨达姆政权的政府发言人如出一辙！	1.2886
No.63	和当年萨达姆政权的政府发言人如出一辙！	0.905
No.151	民主是用导弹和战机换来的吗？民主是用导弹和战机换来的吗？民主是用导弹和战机换来的吗？	0.8772

From the Table 1 above, we can find the opinion leader may change over the time. And opinion leader's ranking is influenced by time. That's the reason why we introduce the feature of time in our proposed algorithms.

In order to evaluate the performance of our proposed Dynamic OpinionRank, we construct a standard by letting expert divide all comments of each period into two categories: strong influence and weak influence. We compare our method with published time, Degree Centrality, Proximity Prestige and OpinionRank proposed in [12] by calculating *F-Score*. The comparison result is shown in Fig.7 below.

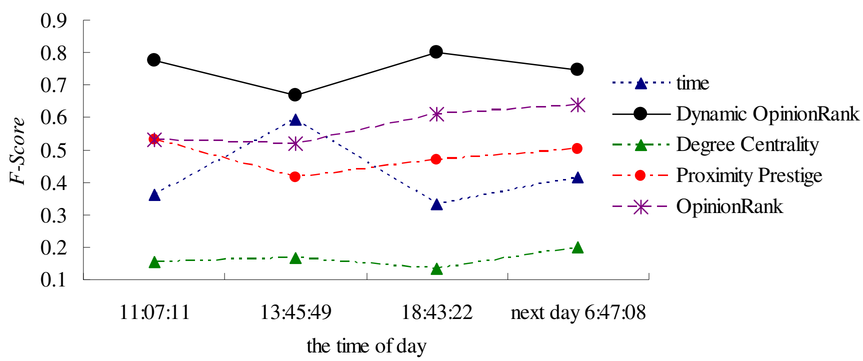


Fig. 7. F-Score of Each Method

From Fig.7, we can find Dynamic OpinionRank has a high accuracy and stability compared with other methods, and it proves the effectiveness of our proposed method.

6.2 Opinion Leader Detection from Users

We apply the method proposed in Section 5.2 for building $G_{UN}(V, E)$ and detecting most influential user. For DBSCAN clustering algorithm, we set radius ranges from

0.06 to 0.12, and initialize MinPts to 1 so that some clusters may contain one single user, which means an outlier. We generally get three to five clusters, and clustering result is shown in Table 2 below.

Table 2. Clustering Results for Detection Opinion Leader from Users

08:04:37~11:07:11		08:04:07~13:45:49		08:04:37~18:43:22		08:04:37~6:47:08	
Cluster	User Count	Cluster	User Count	Cluster	User Count	Cluster	User Count
1	43	1	81	1	117	1	129
2	1	2	1	2	3	2	1
3	2	3	1	3	1	3	1
4	1	4	1				
		5	1				
uid	1220398631	**uid**	1151164004	**uid**	1199491660	**uid**	1725406570

From Table 2, we can remove the first cluster from every cluster group, and use other clusters with fewer points to get the most influential user as opinion leader by formula (9) in each period of time. As $G_{UN}(V, E)$ is very sparse, so we set w_{PP}=10, w_{CQ}=0.3, and w_{DC}=10. At last, we can get opinion leader of each period of time which is shown in the last row of Table 2. According to the experiment, we detect both the most influential comment and user as opinion leader, we also draw a conclusion that the opinion leader may change dynamically as the time goes.

7 Conclusions and Future Work

In this paper, we focus on Chinese news comments, and utilize the techniques such as sentiment orientation analysis, and opinion mining for modeling comment network with explicit and implicit links, and further generating user network. Based on the comment network, we propose an algorithm called Dynamic OpinionRank, for detecting the most influential comment as opinion leader in Chinese news comments. Different from related work, the proposed comment network model considers not only explicit but implicit links, and Dynamic OpinionRank algorithm can detect the most influential comment and track their changes over the time. Moreover, we detect the most influential user in the user network by clustering algorithm according to internal relations between comment and user, and the ranking scores of the comments.

Although this paper puts forward many actual effective methods, there are still some places needed in the later work to improve. As the implicit relationship beyond the context content is difficult to define, there is still no way to find their connections effectively. In addition, combining emotional analysis and natural language grammar characteristics may improve the final accuracy greatly. The other implicit influential features need to be detected. All of them are our future research topics.

References

1. Brin, S., Page, L.: The Anatomy of A Large-Scale Hypertextual Web Search Engine. In: WWW-7 (1998)
2. Dong, Z., Dong, Q.: HowNet (2003), http://www.keenage.com/html/e_index.html
3. Ester, M., Kriegel, H., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: KDD 1996, pp. 226–231 (1996)
4. Feng, L., Timon, C.: Who is talking? An ontology-based opinion leader identification framework for word-of-mouth marketing in online social blogs. Decision Support Systems (DSS) 51(1), 190–197 (2011)
5. Freimut, B., Carolin, K.: Detecting Opinion Leaders and Trends in Online Communities. In: ICDS 2010, pp. 124–129 (2010)
6. Freimut, B., Carolin, K.: Detecting opinion leaders and trends in online social networks. In: CIKM-SWSM, pp. 65–68 (2009)
7. Golaxy: ICTCLAS, http://www.ictclas.org
8. Liu, B.: Web Data Mining: Exploring, Hyperlinks, Contents, and Usage Data. Springer, Heidelberg (2007)
9. Yu, X., Wei, X., Lin, X.: Algorithms of BBS Opinion Leader Mining Based on Sentiment Analysis. In: Wang, F.L., Gong, Z., Luo, X., Lei, J. (eds.) WISM 2010. LNCS, vol. 6318, pp. 360–369. Springer, Heidelberg (2010)
10. Xiao, Y., Xia, L.: Understanding opinion leaders in bulletin board systems: Structures and algorithms. In: LCN 2010, pp. 1062–1067 (2010)
11. Zhai, Z., Xu, H., Jia, P.: Identifying Opinion Leaders in BBS. In: Web Intelligence/IAT Workshops 2008, pp. 398–401 (2008)
12. Zhou, H., Zeng, D., Zhang, C.: Finding leaders from opinion networks. In: ISI 2009, pp. 266–268 (2009)