# A Cluster-Based Opinion Leader Discovery in Social Network

Yi-Cheng Chen, Ju-Ying Cheng and Hui-Huang Hsu
*Department of Computer Science and Information Engineering*
*Tamkang University, New Taipei City*
*Taiwan*
ycchen@mail.tku.edu.tw 602410176@s02.tku.edu.tw h_hsu@mail.tku.edu.tw

***Abstract*—Recently, opinion leader discovery has drawn much attention due to its widespread applicability. By identifying the opinion leader, companies or governments can manipulate the selling or guiding public opinion, respectively. However, mining opinion leader is a challenge task because of the complexity of processing social graph and analyzing leadership quality. In this study, a novel method, TCOL-Miner, is proposed to efficiently find the opinion leaders from a huge social network. We integrate the clustering and semantic analysis methods with some pruning strategies to tackle the influence overlapping issue and the potential leadership of individuals. The experimental results show that the proposed TCOL-Miner can effectively discover the influenced opinion leaders in different real social networks with efficiency.**

***Keywords- clustering, opinion Leader, semantic analysis, social network***

## I. Introduction

Due to the popularity of Web 2.0, social network plays an important role in our daily life. People could share their thoughts and view others' opinions easily. In a social network, opinion leader means the influenced person who may be an expert in a specific domain or have lots of people following his/her comments or ideas. Obviously, opinion leaders are usually the information generator and message senders who familiar with the media by a secondary transmission. The first discussion of this issue is proposed by Katz [13] utilizing the two-step flow of communication. Author points out several characteristics of opinion leader, which involve not only influential power but also influential commentary in a specific domain. Notice that opinion leaders are usually domain-limited; the person who is an expert in one field maybe do not have any follower in other fields.

Clearly, finding opinion leader in a social network has great commercial and political values. By identifying the most influential people, companies or governments can contact them to manipulate selling or guiding public opinion, respectively. Additionally, detecting the most influential comments is also able to understand the source of public opinion formation. For example, to plan for a marketing campaign of a new product via social-networking media, a company may want to target a small number of opinion leaders for a trial of the product, hoping that they could influence their followers, who may in turn influence their friends, to buy the product (or become familiar with it). For consumers and businesses, opinion leaders' ideas or comments are referenced by most users in the network.

A considerable amount of research effort has been put forth on the opinion leader mining [2, 3, 5, 19, 20, 21, 27, 32, 33, 34]. Several studies utilize the characteristics to find opinion leader [5, 19, 20], such as, the number of posted articles, the number of articles replied by others, the number of followers, to name a few. However, these methods may suffer the influence overlapping problem because of ignoring the network structure. The discovered opinion leaders may have many common followers, hence, the influence only can affect a small set of people. Several methods [2, 3, 21, 27, 32, 33, 34] discover opinion leaders by observing the position in the network. Nevertheless, the analysis of network structure is usually time-consuming, especially for large social graphs.

In this paper, a novel algorithm, two-stage cluster-based opinion leader miner (abbreviated as TCOL-Miner), is proposed to efficiently discover opinion leader in a social network. The contributions of our work are as follows,

— Different to previous studies, TCOL-Miner utilizes the post-and-follow relationships among individuals to quickly construct a social network. The time of user writing a post is also considered to adjust the weight of edges in the network.

— To tackle the influence overlapping problem, TCOL-Miner modifies the clustering algorithm, H_Cluster [4] to effectively and efficiently discover the community structure of social network without any parameter setting.

— By the observation, in a social network, all opinion leaders have some common characteristics. TCOL-Miner utilizes *k*-mean clustering algorithm to fast exam the leadership quality of each node and shrink the candidate set by effectively filtering out un-promising nodes.

— Finally, TCOL-Miner finds the tendencies of discovered opinion leaders by analyzing the sentiments and semantics from the content of posted articles. We could correctly identify the positive and negative opinion leader.

— We conduct a comprehensive evaluation by experimentation using real datasets. The experimental results show that algorithm TCOL-Miner significantly outperforms the state-of-the-art algorithms in terms of both efficiency and the influence spread of discovered opinion leader.

The rest of this paper is organized as follows. Section 2 describes some relevant works. Section 3 details the TCOL-Miner framework and associated algorithms. Section 4 presents the experiments on several real datasets. Finally, we conclude in Section 5.

## II. Related Works

### 2.1 Opinion Analysis And Mining

Several studies focus on analyzing and mining opinion in a social network. Liu et al. [18] define the opinion mining problem and discuss the sentiment analysis by assessing the utility and quality of online reviews. An opinion Observer system [17] could analyze and compare the consumer opinions among competing products. Opinion Observer also uses language pattern mining method to extract product features from the Pros and Cons in a specific type of reviews. CopeOpi [15] is an opinion mining system which allows users to input their looking target and to choose the source and time period of information contents to be processed.

Actually, when mining opinion or analyzing sentiment, many thesauruses are generally utilized for measuring the orientation or the property of comment, such as WordNet [25] and General Inquire [10] for English lexicon, HowNet [12] and NTUSD [14] for Chinese lexicon. A proper thesaurus is very important for deciding the users' comments. Ku et al. [14] utilize NTUSD to determine a web comment positive or negative.

### 2.2 Community Deteciton

A community is characterized as a subset of individuals who interact with each other more frequently than other individuals outside the community [29]. Community discovery is similar but not equivalent to the conventional graph partitioning problem. Both community discovery and the graph partitioning problem aim to cluster vertices into groups. A key challenge for the former, however, is to decide what the "most natural" partition of a network is, that is, we do not need to give any heuristic information to guide the partition. Moreover, if there exists no good community structure, there is no need to partition the network. This is why we use the community detection algorithm rather than the graph partitioning algorithm in our research.

A quantitative measure, called modularity, has been proposed [28] to assess the quality of community structures and to formulate community discovery as an optimization problem. Since optimizing modularity is an NP-problem, several heuristic methods have been proposed, as surveyed in Danon et al. [6]. Assume that $n$ is the number of nodes in a social network. The time complexity of most community detection algorithms [1, 6, 7, 8, 9, 11, 16, 22, 23, 24, 26, 28, 30, 31] is between $O(n \log n)$ and $O(n^3)$.

### 2.3 Opinion Leader Mining

Several studies focus on the opinion leader discovery. Zhai et al. [34] propose two methods, interest-field-based and global-measure-based algorithms, to identify opinion leaders in BBS. Miao et al. [21] point out that opinion leaders usually have different expertise and interests, i.e., opinion leaders are deomain-sensitive. We could consider some characteristics, such as expertise, interesting, number of followers, to detect opinion leaders. Several algorithms [19, 20, 5] use different characteristics of user of network to detect opinion leaders. Li et al. [19] develop a framework to identify opinion leaders using the information retrieved from blog content, authors, readers, and relationships. Li et al. [20] propose a mix framework for opinion leader identification in online learning communities. Authors rank opinion leaders based on four distinguishing features: expertise, novelty, influence, and activity. Duan et al. [5] combine clustering algorithm and sentiment analysis to find opinion leader. Features for clustering are calculated based on messages from user activities.

There are many papers modify PageRank algorithm to find opinion leader, such as OpinionRank [33], LeaderRank [32] and Dynamic OpinionRank [27]. Bodendorf et al. [2] present an approach based on social network analysis to detecting opinion trends and leaders. Cho et al. [3] use a threshold model to investigate which opinion leader is the best marketing choice in terms of diffusion speed and maximum cumulative number of adopters.

## III. Proposed Algorithm: TCOL-Miner

Figure 1 shows the framework of TCOL-Miner, which has three components: 1) network construction, 2) opinion leader detection, and 3) sentiment analysis.

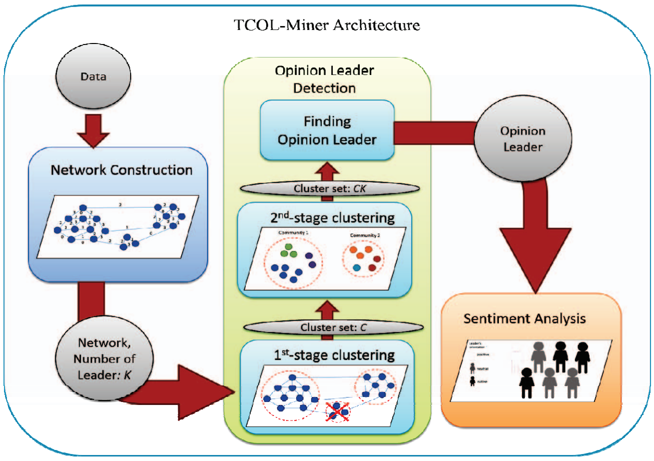


Fig. 1: TCOL-Miner Architecture.

TCOL-Miner utilizes the post-and-follow relationships among individuals to quickly construct a social network. When calculating the similarity (weight) of edge, we also consider the average time of posted article to adjust the similarities among individuals. For detecting opinion leader, we use 1st-stage clustering to efficiently find out qualified community and avoid the influence overlapping problem. Based on the discovered communities, we use 2nd-stage clustering to significantly shrink the candidate size of opinion leaders. Then, TCOL-Miner selects the $k$ best-quality users from the candidate set according to the measuring function. Finally, we propose an efficient sentiment analysis method to detect the discovered opinion leaders positive or negative based on thesaurus NTUSD [14]. Now we detail each component as follows.

### 3.1 Social Network Construction

First, TCOL-Miner evaluates the average post time of articles, $avg\_time_u$, of each user $u$. When calculating the $avg\_time_u$, we must consider both the time of published articles by $u$ and the time of $u$ replied other users' articles. The derived *avg_time* can be treated as the main activity time of user. We segment 24 hours into four sections, 9:00~14:59, 15:00~20:59, 21:00~02:59, 03:00~08:59. TCOL-Miner uses Equation (1) to label each user $u$ a number $T_u$.

$$T_u = \begin{cases} 0, \text{ if } avg\_time = [09:00, 15:00) \\ 1, \text{ if } avg\_time = [15:00, 21:00) \\ 2, \text{ if } avg\_time = [21:00, 03:00) \\ 3, \text{ if } avg\_time = [03:00, 09:00) \end{cases}. \quad (1)$$

Then, to build a social network $G(V, E)$, we treat each user $u$ as a vertex in $G$ and an edge $(u, v)$ exists in $E$ if $u$ and $v$ have reply-article-relation. Suppose the set of adjacent nodes of a node $u$ is defined as $adj(u)$. The similarity $sim(u, v)$ is defined as the common adjacent users of $u$ and $v$,

$$sim(u,v) = \frac{|adj(u) \cap adj(v)|}{\sqrt{|adj(u)| \times |adj(v)|}}. \quad (2)$$

Note that $adj(u)$ also includes u. The weight of edge $(u, v)$ in $V$ is derived by Equation (3),

$$w(u,v) = \frac{sim(u,v)}{|T_u - T_v|}. \quad (3)$$

From above steps, we can construct a social network with weight on the edge.

### 3.2 Opinion Leader Detection

TCOL-Miner discovers opinion leaders in three steps. We use 1st-stage clustering to capture the community structure of social network. Then, we use 2nd-stage clustering to analyze the characteristics of uses and shrink the size of the candidate set. Finally, we pick $k$ users have better leadership quality from candidate set as opinion leaders. We detail each step as follow.

#### a) The 1st-stage Clustering: Community Structure Detection

TCOL-Miner modifies the H_clustering [4] to discover the community structure in social network. Given a social network $G$, we first treat each node as a community and groups each pair of nodes into a community if the structural weight between these two nodes is the largest among their surrounding edges from each other. For example, given two nodes $u$ and $v$, if the edge $(u, v)$ is the largest among all edges connecting to $u$ and also is the largest among all edges connecting to $v$, we merge $u$ and $v$ into a community. Next, we treat each newly created community as a node, and the process continues until a termination condition is reached. TCOL-Miner adopts the modularity gain [4] to measure the quality of discovered communities in order to decide when to stop the community detection process.

TCOL-Miner utilizes the modularity gain as the terminated criteria. At each iteration, based on the clustering result from the last iteration, we merge all pairs of nodes with the strongest structural weight among their neighbors to form larger communities. Suppose the clustering result in the last iteration and in the current iteration are $C$ and $C'$, respectively. If the modularity gain from $C$ to $C'$ is negative, TCOL-Miner will stop clustering, since the previous clustering result is good enough. TCOL-Miner can significantly reduce the time consumed in the clustering. After discovering community structure, we prune some small community with the *size_threshold*. The processing flow is shown in Figure 2.

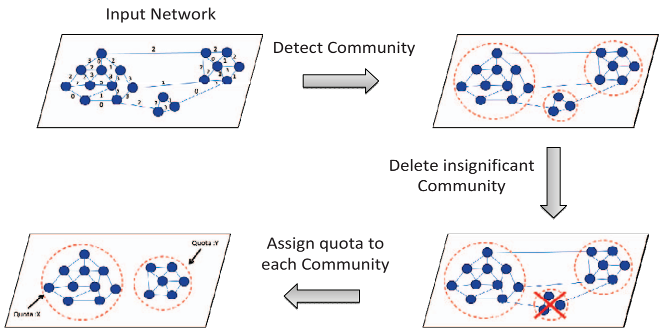


Fig. 2: The concept of 1st-stage clustering in TCOL-Miner.

#### b) The 2nd-stage Clustering: Candidate of Opinion Leader

TCOL-Miner borrows the idea of Duan et al. [5] to find the opinion leaders. Different to [5], we use *k*mean clustering to build the candidate set, and then pick the opinion leaders. With proper selected attributes, the *k*mean clustering can effectively shrink the size of candidate set. TCOL-Miner utilizes four the attributes, *article_num*, *replied_by_prob*, *expert_deg*, and *reply_prob*, derived from users' characteristics to cluster users in a social network, as shown in Table 1.

Table 1: Four attributes for 2nd-stage clustering in TCOL-Miner

| attribute | description |
|---|---|
| ***article_num*** | Total number of articles published by a user |
| ***replied_by_prob*** | The probability of a user's articles replied by other users |
| ***expert_deg*** | The expert degree of a user (total number of articles) in a specific domain |
| ***reply_prob*** | The probability of articles which $u$ reply other users. |

We detail each attribute as follow. For a user $u$, the *article_num* means the total number of articles published by $u$ (including the article posted by $u$ and the article that $u$ replies other users' posts). The *replied_by_prob* means probability of $u$'s articles replied by other users. The *expert_deg* indicates the expert degree of $u$ (total number of articles) in a specific domain. As aforementioned description, opinion leaders are usually domain-sensitive; the person who is an expert in one field maybe have no any follower in other fields. The *expert_deg* means probability of $u$'s articles published in one domain. By our observation, an opinion leader interacts other users frequently in a social network. The *reply_prob* means probability of articles which *u* reply other users.

With the four attributes, TCOL-Miner utilizes *k*mean to cluster users in the significant community discovered from 1st-stage clustering. As shown in Fig. 3, we collect all clusters of each community as the candidate set for opinion leader discovery in next step.

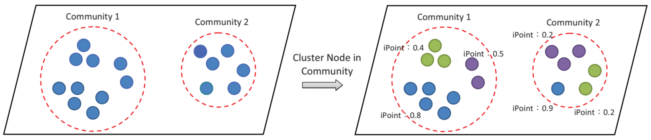


Fig. 3: 2nd-stage clustering in TCOL-Miners.

*c) Opinion Leader Selection*

Now we score each cluster in candidate set and pick the *k* users in the high-score clusters as the final opinion leaders. Given the clustering results $C = \{c_1, c_2, ..., c_n\}$, the evaluation function is defined as Equation (4).

$$score(c_i) = sin(\frac{1}{|c_i|}\sum_{j=1}^{|c_i|} article\_num_j) \times \frac{1}{|c_i|}\sum_{j=1}^{|c_i|} expert\_deg_j$$
$$\times \frac{1}{|c_i|}\sum_{j=1}^{|c_i|} replied\_by\_prob_j \times \frac{1}{|c_i|}\sum_{j=1}^{|c_i|} reply\_prob_j \quad (4)$$

After sorting the clusters by score, we select opinion leaders from each high-score cluster until the number of opinion leaders has reached the specified number *k*.

### 3.3 Sentiment Analysis

The concept of sentiment analysis is shown in Fig. 4. First, we collect the articles published by each discovered opinion leaders. Then, TCOL-Miner analyzes each article. We use NTUSD lexicon [14] to judge the orientation of articles. Since the different keyword may be searched for different fields, we need to modify the content of lexicon. After modifying the lexicon, TCOL-Miner puts positive words into the set *P* and the negative words into the set *N*, and then calculates the numbers of positive and negative words to judge the orientation of the article.

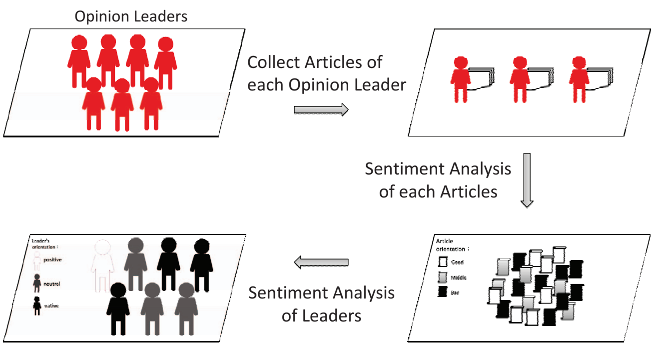


Fig. 4: The concept of Sentiment Analysis.

Finally, with the result got from the last step, TCOL-Miner can decide the orientation of each discovered opinion leaders. Given a opinion leader *OL*, If the size of *P* set is larger than *N* set, we could judge *OL* is a positive leader. This type of leaders usually gives some positive or good comments in the social network. On the contrary, if the size of *P* set is less than *N* set, we could judge *OL* is a negative leader. Clearly, this type of leaders usually criticizes something and gives some bad or negative comments in the social network. Actually, when deciding the type of leader, if the difference between the size of *P* and *N* does not exceed a threshold, we usually judge *OL* as a neutral leader.

## IV. EXPERI-MENTAL RESULTS

TCOL-Miner integrates network structure and leadership quality analyzing method to efficiently find opinion leaders in a social network. For performance discussion, we compare the mining algorithm of TCOL-Miner with leadership quality clustering algorithm, att_clustering [5]. All algorithms were implemented in C++ language and tested on a workstation with Intel i7-3370 3.4 GHz with 4 GB main memory. The performance study has been conducted on real world datasets. First, we conduct experiment to observe the influence of two algorithms with varying the number of opinion leaders. Then, we compare the performance and the practicability at different candidate number of two methods.

### 4.1 Real Dataset Collection

The dataset LEXUS and AUDI used in the experiment are the discussion of car collected from Mobile01 Forum [35]. The crawler used to fetch data is implemented in Python language. The content of dataset include the subjects of article, article content, author, reply content, and the time of publish or reply articles. Table 2 lists the information of two real world dataset.

Table 2: Information of LEXUS and AUDI dataset

| Name | Articles | Reply relations | Members |
|---|---|---|---|
| LEXUS | 6928 | 126454 | 23797 |
| AUDI | 6418 | 97835 | 21130 |

### 4.2 Experiments And Discussion

We compare TCOL-Miner with att_clustering which use a clustering method only based on the characteristics of individuals to find opinion leader. To evaluate the results of discovered opinion leaders, we use the influence spread as metric. An individual can spread influence to another individual if there exists a link between each other in a social network. In first experiment, we target the dataset LEXUS and fix the number of desired opinion leaders *n* as 200. As shown in Fig. 5, we can find that the influence spread (potentially influenced users) of opinion leaders discovered by TCOL-Miner is larger than those discovered by att_clustering, with varying the number of *k* in *k*-mean. Obviously, this phenomenon mentions that, in TCOL-Miner, the 1st-stage clustering could significantly solve the influence overlapping problem of opinion leaders in social network. Additionally, as shown in Fig.4, when *k* setting as 2 to 4, TCOL-Miner has the best influence results. However, att_clustering only can achieve the best result under $k = 2$.

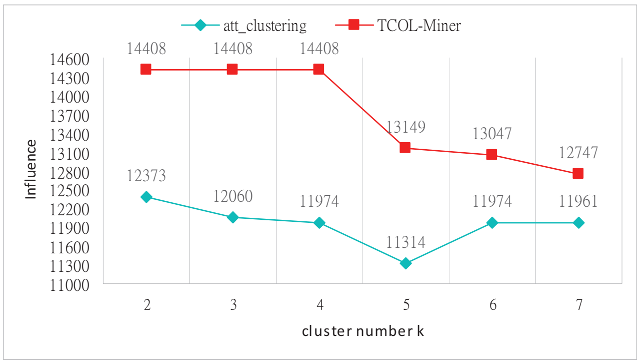


Fig. 5: The influence spread of two algorithms with varying $k$ in $k$-mean

In second experiment, we discuss the influence spread of two algorithms under varying desired number of opinion leaders. According to Fig. 6, in both dataset LEXUS and AUDI, the result of TCOL-Miner is better than att_clustering. By the observation, the influence of att_clustering increase slowdown when n is greater than 130. We can clearly point out that influence overlapping problem is serious in this criterion. The steady increase of influence of TCOL-Miner indicates that the proposed method can effectively reduce the impact of influence overlapping problems and discover opinion leaders with better qualities.


Fig. 6: The influence spread of two algorithms with two real datasets

## V. CONCLUSION

Recently, opinion leader discovery has drawn much attention due to its widespread applicability. In this paper, we develop a novel algorithm, TCOL-Miner, which integrates network structure and leadership quality analysis methods to efficiently find opinion leaders in a large social network. We utilize the two-stage clustering methods to significantly reduce the impact of influence overlapping problem in social network. TCOL-Miner also utilizes the sentiment analysis to distinguish the opinion trend of discovered opinion leaders. Finally, to mention the practicability, we perform the proposed algorithm on real datasets. The experimental results show that TCOL-Miner could detect more qualified opinion leader under different criteria and effectively solve the influence overlapping problem.

## ACKNOWLEDGE


Yi-Cheng Cheng was supported in part by the Ministry of Science and Technology of Taiwan, Project, No. MOST 104-2221-E-032-037-MY2.


## REFERENCES


[1] D. Bortner and J. Han, "Progressive clustering of networks using structure-connected order of traversal," *In Proceedings of the 26th IEEE International Conference on Data Engineering (ICDE'10)*, pp.653–656, 2010.

[2] F. Bodendorf, and C. Kaiser. "Detecting Opinion Leaders and Trends in Online Social Networks," *Proceedings of the 2nd ACM workshop on Social web search and mining*, pp. 65-68, 2009.

[3] Y. Cho, J. Hwang, and D. Lee. "Identification of Effective Opinion Leaders in the Diffusion of Technological Innovation: A Social Network Approach," *Technological Forecasting and Social Change*, vol. 79, no.1, pp. 97-106, 2012.

[4] Y.C Chen, W.Y Zhu, W.C Peng, W.C Lee, and S.Y Lee. "CIM: community-based influence maximization in social networks, "*ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no.2, pp. 25, 2014.

[5] J. Duan, J. Zeng, and B. Luo. "Identification of Opinion Leaders Based on User Clustering and Sentiment Analysis," *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pp. 377-383, 2014.

[6] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech. Theory Exp,* vol. 2005, no. 9, 2005.

[7] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pp.226–231, 1996.

[8] Z. Feng, X. Xu, N. Yuruk, and T. Schweiger, "A novel similarity-based modularity function for graph partitioning," *In Proceedings of the 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'07)*, pp. 385–396, 2007.

[9] M. Girvan and M. Newman, "Community structure in social and biological *networks*," *Proc. Nat. Acad. Sci*, vol. 99, no. 12, pp. 7821–7826, 2002.

[10] General Inquire. http://www.wjh.harvard.edu/~inquirer/.

[11] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, and Y. Liu, "SHRINK: A structural clustering algorithm for detecting hierarchical communities in networks," *In Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM'10),* pp. 219–228, 2010.

[12] HowNet. http://www.keenage.com/html/e_index.html.

[13] E. Katz, "The two-step flow of communication: An up-to-date report on an hypothesis." *Public opinion quarterly*, vol. 21, no. 1, pp. 61-78, 1957.

[14] L.W Ku, and H.H Chen, "Mining Opinions from the Web: Beyond Relevance Retrieval," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 12, pp. 1838-1850, 2007.

[15] L. Ku, H. Ho, and H. Chen, “Opinion Mining and Relationship Discovery using CopeOpi Opinion Analysis System,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 7, pp. 1486-14503, 2009.
[16] A. Lancichinetti, S. Fortnato, and J. Keryesz, “Detecting the overlapping and hierarchical community structure in complex network,” *New J. Physics*, vol. 11, no. 3, 2009.
[17] B. Liu, M. Hu, and J. Cheng, “Opinion Observer: Analyzing and Comparing Opinions on the Web,” *Proceedings of the 14th international conference on World Wide Web*, pp. 342-351, 2005.
[18] B. Liu, and L. Zhang. “A Survey of Opinion Mining and Sentiment Analysis,” *Mining Text Data*, pp. 415-463, 2012.
[19] F. Li, and T. Du, “Who is Talking? an Ontology-Based Opinion Leader Identification Framework for Word-of-Mouth Marketing in Online Social Blogs,” *Decision Support Systems*, vol. 51, no. 1, pp. 190-197, 2011.
[20] Y. Li, S. Ma, Y. Zhang, R. Huang, “An Improved Mix Framework for Opinion Leader Identification in Online Learning Communities,” *Knowledge-Based Systems*, vol. 43, pp. 43-51, 2013.
[21] Q. Miao, S. Zhang, Y. Meng, and H. Yu, “Domain-Sensitive Opinion Leader Mining from Online Review Communities,” *Proceedings of the 22nd international conference on World Wide Web companion*, pp. 187-188, 2013.
[22] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” *In Proceedings of the Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS’01)*, pp. 849–856, 2001.
[23] M. Newman, “Fast algorithm for detecting community structure in networks,” *Phy. Rev. E*, vol.69, no.6, 2004.
[24] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the *overlapping* community structure of complex networks in nature and society,” *Nature*, vol. 435, no.7043, pp. 814–818, 2005.
[25] Princeton University “About WordNet.” WordNet. Princeton University. 2010. http://wordnet.princeton.edu.
[26] J. Ruan and W. Zhang, “An efficient spectral algorithm for network community discovery and its applications to biological and social networks,” *In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM’07)*, pp. 643–648, 2007.
[27] K. Song, D. Wang, S. Feng, and G. Yu, “Detecting Opinion Leader Dynamically in Chinese News Comments,” *Web-Age Information Management.Springer*, pp. 197-209, 2012.
[28] L.Wan, J. Liao, and X. Zhu, “Finding evaluating community structure in social networks,” *In Proceedings of the 4th International Conference on Advanced Data Mining and Applications (ADMA’08)*, pp. 620–627, 2008.
[29] S.Wasserman and K. Faust, “Social network analysis: Methods and applications,” *Cambridge University Press*, 1994.
[30] S. White and P. Smyth, “A spectral clustering approach to finding communities in graph,” *In Proceedings of the 5th SIAM International Conference on Data Mining (SDM’05)*, pp. 274–286, 2005.
[31] X. Xu, N. Yuruk, Z. Feng, and T. Schweiger, “SCAN: A structural clustering algorithm for networks,” *In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’07)*, pp. 824–833, 2007.
[32] X. Yu, X. Wei, and X. Lin. “Algorithms of BBS Opinion Leader Mining Based on Sentiment Analysis,” *Web Information Systems and Mining.Springer*, pp. 360-369, 2010.
[33] H. Zhou, D. Zeng, and C. Zhang. “Finding Leaders from Opinion Networks,” *Intelligence and Security Informatics, 2009. ISI'09.*, pp. 266-268, 2009.
[34] Z. Zhai, H. Xu, and P. Jia. “Identifying Opinion Leaders in BBS,” *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08.*, vol. 3, pp.398-401, 2008.
[35] Mobile01. http://www.mobile01.com