

ITC 4/48 Information Technology and Control Vol. 48 / No. 4 / 2019 pp. 522-537 DOI 10.5755/j01.itc.48.4.21950	<b>Mining Hot-Personae Approach Based on Local Social Microblog Graph</b>	
	Received 2019/04/22	Accepted after revision 2019/10/09
	 <a href="http://dx.doi.org/10.5755/j01.itc.48.4.21950">http://dx.doi.org/10.5755/j01.itc.48.4.21950</a>	

# Mining Hot-Personae Approach Based on Local Social Microblog Graph

**YaJun Du, Biao Peng**

School of Computer and Software Engineering, Xihua University, Chengdu 610039, China;  
phone: +86 87720157; fax: +86 87726572; e-mail: duyajun@mail.xhu.edu.cn

**FangHong Su, Fei Chen**

Sichuan Lewei Technology Co., Ltd., Chengdu 610041;  
phone: +86 18109012878; e-mail: 1977346279@qq.com

**ShangYi Du**

ChengDu No. 7 High School, Chengdu 610000, No. 1 Fubei Road, Jiannan Avenue, Chengdu;  
e-mail: 18200204019.qq.com

---

Corresponding author: 1977346279@qq.com

---

With the increasing popularity of online social media platforms, netizens always chat with their friends and share information, such as what they like in their daily lives, on these platforms. Netizens publish tons of information on social platforms every day. These platforms converge many people and information. The processes by which the publishers find the sharers who are interested in their publications and the sharers find some interesting things and information in what the publishers published have resulted in the challenge of retrieving information from social network fields. To address these issues, we propose a novel algorithm, named Hot Persona Mining, to analyze the users' focus personae from microblog posts in the online social networks. During mining, we first utilize local-based graph clustering to establish the nearest neighbor nodes of target users. Then, we mine users' focused personae entities from their neighbors' published microblog posts in different periods. Then, we construct the users' active score vector and their interest matrix to mine the hot personae in every local social graph. The experimental results show that our algorithm effectively mines current focus of the target user, and exhibits good performance as shown by its precision, recall and F-measures.

**KEYWORDS:** Microblog, focus mining, graph clustering, online social network.

## 1. Introduction

The Internet Ecosystem has emerged because of the evolution of the Internet Plus Age. In such an environment, each netizen acts like an active cell living in the internet. Similar to living in the real world, netizens always communicate with their friends in online social platforms (e.g., Tencent Microblog, RenRen, Wechat, Facebook, and Twitter are the most popular). In the initial stages of Twitter, the posts increased at rate of 200,000 per day [31]. Netizens have formed many huge online social networks [26, 34, 36]. In such a huge and complex relationship, that is, the social network system, the effective mining and recommending of users' information [15] remain a great challenge for current studies, especially in the absence of network information [10]. One or several posts in microblog platforms usually cannot offer useful information. However, a segment post within a time sequence can conceal many bits of valuable information and knowledge that can be useful to various people, such as the public and the government.

Two recent aspect studies based on link structure graphs [19] and microblog semantic context [11] are used to recommend followees, followers, interest friends, influence persons, and others.

### 1.1. Link-Structure-Based Recommending Approaches

On the one hand, constructing the netizens' online social graphs and mining their focal areas from their internet behavior are very important for user profiling. Graph mining approach based on the classic graph theory is a relatively new area of research in online social network microblog platform [28]. Yu et al. [44] showed that traditional data mining methods are not suitable for online social networks that do not provide users' preferences or rating data. However, link prediction is superior to other methods in online social networks with sparse user characteristics, so this method is used to extract the missing information in many cases [25]. Link prediction methods based on graph theory are usually used to predict whether user will make a friend with anybody in the future. The computation using link prediction only needs to consider the users' link relations, and detailed features are not required. Only the similarities among the nodes in the graphs should be considered [25, 30].

Chen et al. [6] proposed the Friend of a Friend algorithm (FOFA) based on common nodes, that is, the more similarities between two nodes, the more possibilities that the users will establish a direct link in the future. Leicht et al. [18] proposed hub depressed index (LHNI) algorithm to measure the similarity of two nodes. Hence, the similarity of two nodes rest with their neighbors' similarity. Adamic et al. [1] proposed the Adamic/Adar index (AA) based on the multi-feature count. In computing AA, the rarer the nodes' common features, the larger the weight will be given. Dao et al. [9] computed the similarity of two elements of a database according to the Markov diffusion and the commute time kernels. Tong et al. [38] proposed the Random Walk with Restart (RWR) algorithm. Its basic ideal is based on the Markov chain mode. They mainly consider the linear correlation and block-wise, community-like structure in the graph. Papadimitriou et al. [30] designed a novel link prediction algorithm, named FriendLink (FL), to compute for the similarity of two nodes according to the number of path between two nodes and depth of nodes. Peng et al. [32] proposed the Active-friend Prediction (AFP) approach. They regard each user as a basic unit, named as target user, and predict who will become the target user's friend in the future. AFP considered the "ratio of nodes" out-degree to in-degree as the weights for measuring the nodes' activity. Zhao et al. [45] proposed a user influence rank (UIRank) algorithm to identify the influential users by considering the contribution of user's tweet and follower graph. Yao et al. [41] considered the following rate (FR) factor, activity (ACT) factor, authority (ATR) factor, interaction (ITA) factor and similarity (SML) factor of social networks, proposed an identifying influential user algorithm by using Support Vector Regression (SVR) model.

### 1.2. Content-Feature-Based Recommending Approaches

Netizens frequently share posts of what they are interested in. The content of these posts, which is mentioned in the published posts, always describes their life circles and hobbies [14, 37]. Thus, extracting named entity from posts is the key to analyze users' interest. Jung et al. [16] proposed a novel Named En-

tity Recognition (NER) system based on microtext clustering method. The system is suitable for the streaming texts. Yu et al. [43] proposed the Chinese named entity identification approach based on the cascaded hidden Markov model (CHMM). The proposed approach can effectively recognize a person's name, location name, and organization name from large realistic corpus. Li et al. [20] proposed a two-phase strategy based on CRFs. Their method is more efficient than one-phase method in recognizing Chinese name entities. Fan et al. [12] proposed semantic spreading model to extract the users' implicit interests from posts by exploiting Wikipedia knowledge. Nicoletti et al. [29] proposed an unsupervised method associated with Wikipedia articles to mine interests for user profiling. Peng et al. [33] proposed Focusing Personae Extraction (FPE) method to extract implicit and explicit focuses from the target user's high-similarity social network.

Previous studies did not consider the combination of the structural features of microblog platforms and the content features of the posts. Liu et al. [21] concentrated on both community structure and micro-blog content, proposed a detecting social community algorithm to find all users in social community.

In this paper, we focus on deep mining for hot personae to find the latent people who owned the most netizens' reviews on the local social graph. The main contributions of studies embody the following aspects:

- Given the lack of user attribute information in microblog platforms, we abstract the netizens (nodes) and their reviews for each other (edges) into the digraph. We divide the users into active, inactive and authoritative users, and define the active ratios of the users by considering followers and followees. Based on the link (the review relationships among netizens) prediction, we propose the connection degree algorithm by considering the direct and indirect review relationship.
- We work out a users' activity score by combining the user's connection degrees with the users' active ratios. Furthermore, we propose the algorithm to construct the users' highly active Local Social Graph (LSG) by extracting the active indirect or direct neighbors who have potential relationship with the user according to an active score.

- The microposts with short texts in the microblog platform contain personae that are the focal personae. We extract personae from micro posts published by the users in his/her LSG. Inspired by the locally weighted regression (LER) [8], we assign a higher weigh to recent published post, and define the attention rate for personae. We propose the algorithm of mining implicit and explicit focusing personae by regarding the personae with higher attention rates as focal personae.
- Finally, we compare our method with other link-based similarity approaches using precision, recall, and F-score. The results show that our hot persona mining (HPM) algorithm has higher precision, recall, and F-measure and more stable than other link similarity measures. The proposed algorithm can effectively extract the user's focal personae.

The rests of this paper are organized as follows. Section 2 introduces some link similarity measures and techniques for mining user interests. This section also provides an abstraction of the social network. Section 3 describes our method in detail. Section 4 evaluates the precision, recall, and F-measure of our algorithm using real data sets crawled from the online social network. Section 5 presents the conclusions and future works.

## 2. Related Work

In this section, we detail some users' focus mining methods and link similarity methods.

### 2.1. Focus Mining Methods

Current studies on the users' focus mining attempt to extract the users' interests from online social network. Nicoletti et al. [29] mined users' interests from their electronic conversations in 2013. Their method contain the following steps:

**Step 1.** Message and user detection. Parse users' electronic conversation to group these messages by users and remove irrelevant contents.

**Step 2.** Message pre-processing. Noisy texts (such as stop-word and invalid characters) are filtered out.

**Step 3.** Concept association. A semantic dictionary is built, and it contains the concept of human knowledge associated with the texts.

**Step 4.** Category hierarchy generation. Concepts are extracted from semantic dictionary, and these concepts form a hierarchical structure.

Each hierarchical structure construction represents a user profile and a lower level indicates higher relevancy with the user.

Liu et al. [23] proposed keyword-based method to mine the microbloggers' interests from their posts. They download the users' posts from online social network according to the user ID and then remove irrelevant information, such as repost messages, mentioned names, emotion icon, and URLs. After cleaning the data, they divide the post and tag part of the speech. Finally, they extract and combine keywords. Lu et al. [24] combined four measure indices, namely, page rank, betweenness centrality, closeness centrality, and out-degree, and proposed ranking influential users' algorithms. Peng et al. [32] proposed the FPE method to mine the target users' focal areas from the published posts in his/her LSN. In this paper, the FPE first constructs the users' LSN according to the similarity measure between two nodes and lists the top  $k$  users to construct the similarity vector  $\vec{S} = (s_1, s_2, \dots, s_k)$  according to the similarity, where  $s_i$  is the  $i$ th users' similarity. Then, the FPE download the published posts of the top  $k$  users and extract persona entity (PE) to construct the user-PE matrix  $W$  from these posts. Each element  $W_{ij}$  in matrix  $W$  represents the number of  $i$ th user's recent published posts which mentioned the  $j$ th PE. Finally, they calculate the attention rate of the PEs using the product of Sand  $W(S \times W)$ . In summary, FPE filters out users with lower similarity, and it only needs to mine user's focal areas from his/her LSN. Hence, the FPE is less complex than the direct focus mining method.

## 2.2. Link Similarity Methods

Furthermore, the experiments in these papers [2, 27, 32] show that the similarity measure has great influence on the effect of focus mining. Thus, choosing an appropriate similarity measure is also the key to improve the precision and recall of FPE. In the following passage, we elaborate several approaches to measure the similarity of nodes.

Papadimitriou et al. [30] proposed the local-based FriendLink (FL) algorithm and considered the length of pathways. The FL relies on the paths from  $v_a$  to  $v_b$  and the length of these paths. Thus, the similarity be-

tween the two nodes is computed by Equation (1):

$$Sim(v_a, v_b) = \sum_{i=2}^l \left( \frac{1}{i-1} \cdot \frac{|path(s_{v_a, v_b}^i)|}{\prod_{j=2}^i (n-j)} \right), \quad (1)$$

where  $n$  is the number of total nodes.  $|path(s_{v_a, v_b}^i)|$  is the number of all length- $i$  paths from  $v_a$  to  $v_b$ .  $l$  is the maximum length between  $v_a$  and  $v_b$ .

Chen et al. [6] proposed the well-known Friend of a Friend algorithm (FOFA). Their paper indicates that the more common friends two users have, the more likely they will become friends in the future. The FOFA considers the number of common friends of  $v_a$  and  $v_b$ , and their similarity formula is shown by Equation (2):

$$sim(v_a, v_b) = |\Gamma(v_a) \cap \Gamma(v_b)|, \quad (2)$$

where  $\Gamma(v)$  indicates the set of neighbors of node  $v$ .

Leicht et al. [18] proposed the LHNI algorithm based on the self-similar of the immediate neighbors of two nodes. The similarity between nodes  $v_a$  and  $v_b$  can be computed by Equation (3):  $sim$

$$sim(v_a, v_b) = \frac{|\Gamma(v_a) \cap \Gamma(v_b)|}{|\Gamma(v_a)| \times |\Gamma(v_b)|}, \quad (3)$$

where  $\Gamma(v)$  indicates the set of neighbors of node  $v$ .

Adamic et al. [1] proposed the AA algorithm based on the common-shared features of two nodes. In AA, the rarer feature is given a heavier weight. Its similarity between nodes  $v_a$  and  $v_b$  is defined as Equation (4):

$$sim(v_a, v_b) = \sum_{z \in \Gamma(v_a \cap v_b)} \frac{1}{\log(|\Gamma(z)|)}, \quad (4)$$

where  $\Gamma(v)$  is a set of common neighbors of  $v_a$  and  $v_b$ .

Peng et al. [32] proposed the AFP based on the number of paths from  $v_a$  to  $v_b$ . The ratio of out-degree to in-degree of  $v_b$  also influences their similarity. The more active the users are, the higher probability there is that they will be recommended. The similarity formula is defined as Equation (5):

$$Sim(v_a, v_b) = \left[ \sum_{i=2}^l \frac{1}{i} \times \frac{|path(s_{v_a, v_b}^i)|}{n_i} \right] \times \left( \frac{N_{od\_b} \text{ MinId}}{N_{id\_b} \text{ MaxOd}} \right), \quad (5)$$

where  $n$  is the number of total nodes;  $|path(s_{v_a, v_b}^i)|$  is the number of all length- $i$  paths from  $v_a$  to  $v_b$ ;  $l$  is the maximum length between  $v_a$  and  $v_b$ ;  $N_{od_b}$  and  $N_{id_b}$  are out-degree and in-degree of  $v_b$  in the global social network; and the ratio of  $N_{od_b}/N_{id_b}$  represents the user's activity.  $MinId$  and  $MaxOd$  are the normalizing factors of  $N_{od_b}/N_{id_b}$ .

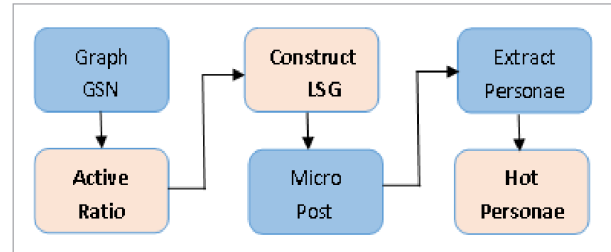
### 3. Hot Persona Mining (HPM) Algorithm

A large amount of link relationships among friends and posts can be found in the microblog platforms. Consequently, the application systems that use these relationships of microblog have been developed. These applications include microblog semantic context retrieval systems [11], top-k followee recommendation over microblog systems [5], expert finding systems using follower relation, user-list relation [39], and hot event detection systems monitoring the cost-effective nodes in microblog platforms [7]. All application systems do not come down to mine and analyse the persona features from the micro posts. These are some differences between our proposed method with FL, FOFA, LHNI, AA algorithms as follows:

- Our proposed HPM algorithm (Figure 1) not only needs to extract the personae, but it also mines some hot personae. Based on global social network (GSN), the HPM algorithm first computes the active ratios of every users. Secondly, the HPM constructs local social network (LSG) by using active ratios. Finally, the HPM algorithm extracts the candidate personae by analyzing the microblog posts on LSG, then recommends the hot personae.
- Our proposed HPM algorithm includes three main models (marked pink boxes in Figure 1), namely, the analysis of active user, construction of the local social graph, and mining hot personae.
- Our proposed HPM algorithm is prominently different from other studies which only focused on the global social network structure.
- We consider whether one person catches very hot attention of the public or not. This characteristic does not mean that all persons of the GSN, but most people in the local public social network pay close attention to the user.

Thus, the HPM focuses on mining the hot personae in the local social network graph, and extracts these hidden personae in micropost contents.

**Figure 1**  
Flow of Our Proposed HPM



#### 3.1. Active Ratio

Several studies explain several indices to measure the importance of the nodes in a microblog platform [19]. They analyze the topological structure of retweet network to find relations of “friendship” at Twitter. To some extent, Yin et al. [42] demonstrate that the numbers of the followees and followers reflect the user features in the microblog platforms. The number of the user followees usually measures the active degrees and the user participation degrees. The more the users' followees there are, the higher the users' actives and participation degrees. By contrast, the users with less followees are not active, and they do not positively participate in the microblog platforms. The number of the user followers usually measures the user influences and the user authorities. The more the users' followers there are, the bigger the users' influences and authorities will be. However, users with fewer followers have smaller influences and lower authorities [22]. In online social network, almost all users have their own social circles. Given that users frequently communicate with their good friends, they share or comment information on what they are interested in with each other. Hence, users' active friends commonly have similar focus with them. The new posts can always reflect users' and their friends' current focus. In such a hybrid online social network [42], the relationships between users A and B have four states as follows:

- A follows B, but B does not follow A;
- B follows A, but A does not follow B;
- A and B follow each other;

Neither A follows B, nor B follows A.

For this reason, we abstract online social network into a digraph to introduce our algorithm. We describe the digraph as  $G = (V, E)$ , where  $V$  represents a set of nodes in this network;  $E$  represents a set of edges in this network,  $e_{ij} = (v_i, v_j)$  represents that node  $v_i$  is the follower of  $v_j$  ( $v_i \rightarrow v_j$ ), where  $v_i \in V$  and  $v_j \in V$  represent persona entities. For the microblog platform, every node of the social graph has enough followees and followers. The proportion  $AR$  (active ratio) of followees to followers indicates the active degrees of the users of nodes.

- $AR < 1$  indicates that the users are active. They are pleased to make friends with other users.
- $AR = 1$  indicates that the users are not active. They are not pleased to make friends with other users.
- $AR > 1$  indicates that the users are authoritative in the microblog platform. Their posts usually affect other users.

$AR$  is considered as the important index in measuring the active degree of the common user  $i$  in the microblog platform. We formalize and normalize the  $AR$  by Equation (6):

$$AR_i = \frac{N_O^i}{N_I^i} * \frac{\text{Min}_{\forall j \in V} N_I^j}{\text{Max}_{\forall j \in V} N_O^j}, \tag{6}$$

where,  $N_O^i$  is the number of the followees of user  $i$ . For the directed graph of microblog platform, the  $N_O^i$  is the out-degree of node  $i$ .  $N_I^j$  is the number of followers of user  $i$ . For the directed graph of microblog platform, the  $N_I^j$  is the in-degree of node  $i$ .  $\text{Min}_{\forall j \in V} N_I^j$  and  $\text{Max}_{\forall j \in V} N_O^j$  are the minimum in-degree and maximum out-degree for all nodes of the directed graph of microblog, respectively.  $\frac{\text{Min}_{\forall j \in V} N_I^j}{\text{Max}_{\forall j \in V} N_O^j}$  is a normalized factor and enforces the  $AR_i \in (0,1]$ .

In this subsection, we discuss the concept on the GSN of online social networks, analysis the following relationship of among users, propose the active ratio of the users.

### 3.2. Construction of the LSG

In a microblog platform, small or large groups (communities) play a major role in deciding the functionalities [4]. The users with close connection are usually clustered into a group. Very few users in the group are praised enormously and become hot personae,

because most other users are interested in some viewpoints in their microposts. These users with this phenomenon are usually visualized into a cluster in the directed graph of the microblog platform. Kim et al. [17] confirmed that academic fields of physics, biology, and education take the central position in co-authorship network by using scholarly article scopus database, respectively. It reasonably explains the phenomenon.

Inspired by FL Algorithm [30], we consider that these nodes in an cluster are closely connected. The connection degrees (Equation (7)) between two nodes  $i$  depend on the number of the achieved paths. If node with higher  $AR_i$  has the connection degrees with its' surrounding nodes, the user of the node is a hot persona.

$$Cd(i, j) = \sum_{k=2}^l \frac{1}{k} \times \frac{|path_{(i,j)}^k|}{n_k}, \tag{7}$$

where,  $Cd(i, j)$  indicates whether users  $i$  and  $j$  are closely connected in microblog platform or not. As  $Cd(i, j)$  becomes bigger, the connection between users  $i$  and  $j$  become more closely connected;  $l$  is the length of the max link path between nodes  $i$  and  $j$ ;  $\frac{1}{k}$  is the impact factor of the length  $k$ . The longer the length of link paths to the node  $i$  is, the smaller the connection degrees to nodes will be;  $|path_{(i,j)}^k|$  is the number of the achieved paths with length  $k$  from node  $i$  to node  $j$ ;  $n_k$  is the number of the achieved paths with length  $k$  from nodes  $i$ .

Otherwise, user  $j$  supports user  $i$  to become an active user in the social network. This process depends on  $Cd(i, j)$  and  $AR_j$ . We synthesize them into  $AfScore(i, j)$  by Equation (8).  $AfScore(i, j)$  indicates that user  $j$  obtains the active score in the local social network around user  $i$ .

$$AfScore(i, j) = AR_j \times Cd(i, j) \tag{8}$$

To mine the focused hot personae and reduce dimension [39] of the  $AfScore(i, j)$ , we have to construct the LSG of the target user from the target users' direct neighbor and high-similarity indirect user. We describe the construction of LSG of the target user. The pseudo-code of the link similarity measure is shown in Algorithm1, and Algorithm2 is the pseudo-code of

obtaining the initial LSG of the target user. Finally, we obtain the LSG of the target user according to the result of Algorithm 1. The LSG of the target user is constructed as follows:

**Step 1.** The target user's ID is encoded;

**Step 2.** The target user's direct neighbor is determined;

**Step 3.** The indirect neighbor is determined according to the direct neighbor list and the back chain in this process;

**Step 4.** The fans (fans of user  $A$  are some users that they like user  $A$  very much and are very interested in user  $A$ ) and idols (idols of user  $A$  are some users that user  $A$  worships and believes them) of each user are counted;

**Step 5.** The relationships between the target user and the neighbors are analyzed;

**Step 6.** The connection degrees between the target user and the neighbors are calculated;

**Step 7.** The users' whole score over a certain threshold is screened out, and these users construct the LSG of the target user.

**Algorithm 1:** LinkSimilarity ( $v, l$ )

```

01 Input
02  $v$  target vertex, it is in level 0;
03  $l$  most effective level, it is limited;
04 GSN  $G$ ;
05 Output
06  $Cd(i, j)$  between nodes  $i$  and  $j$  in  $G$ ;
07 Begin
08 get  $MinId$  and  $MaxOd$  from GSN  $G$  of target user;
09 For  $i = 1$  to  $N$ //traverse all the users in LSG;
10  $G'_i = \text{ConstructLSG}(i, l)$ //construct LSG of  $i$ ;
11 DeleteUnreasonableChainsFrom  $G'_i$ ;
12  $n_i$  = the number of users of  $G'_i$ ;
13  $n_j$  = the number users of level  $l$ ;
14 For  $k = 2$  to  $l$ 
15     For  $j = 1$  to  $n_j$ // traverse users in current level;
16         calculate  $Cd(v_i, v_j)$  between nodes  $v_i$  and  $v_j$ ;
17     End For  $j$ ;
18 End For  $k$ ;
19 End For  $i$ ;
20 End.
```

In algorithm 1, Step 08 computes the minimum in-degree  $MinId$  and maximum out-degree  $MaxOd$  for all nodes of GSN of target user. Steps 14~18 compute the connection degrees of the user  $i$  by traversing all nodes except the user  $i$ .

**Algorithm 2:** ConstructLSG ( $v, l$ )

```

01 Input
02  $V$ : target vertex in level 0, and defined
   by vertexInfo;
03 GSN  $G$ ;
04 Output
05  $G'$ :  $v$ 's LSG;
06 Begin
07  $v_0 \leftarrow v$ ;
08  $G' \leftarrow Null$ ;
09  $Q_0$ .Enqueue( $v_0$ ) //  $Q_0$  is a queue;
10 For  $m \leftarrow 0$  to  $l$ 
11 Begin
12     While  $Q_m$  is not empty;
13     Begin
14          $v_m \leftarrow Q_m$ .Dequeue
15         For each vertex.id of  $G$  in  $v_m$ .list
16         Begin
17             If( $V$  not in  $Q_{m-1} \wedge \text{vertex.l-v}_m=1$ ) //this
               condition prevents back chains,  $v_l$  represent
               the level of vertex's first appearance;
18             Then
19                  $Q_{m+1}$ .Enqueue(vertex)
20                 If(vertex not in  $G'$ );
21                 Begin
22                      $G'$ .Add(vertex);
23                 EndIf
24                  $G'$ .Link( $v_m$ , vertex);
25             EndIf
26         End For;
27     End While;
28 End For;
29 Return  $G'$ ;
30 End
```

In algorithm 2, Steps 10~28 process all nodes which keep the link path length  $0, 1, \dots, l$  with  $v$ . We construct an enqueue  $Q_m$  for each link path length  $m$  to save all nodes of link path length  $m$ . Steps 12~27 process all nodes which keep the link path length  $m$  with  $v$ . Step 17 avoids back-chains.  $v.l$  represents the link path length of  $v$  with target node. Step 19 places nodes of different levels into their corresponding enqueues. Step 22 puts the current node into the local social network graph of  $v$ . Step 24 establishes the directed

links between nodes with link similarities.

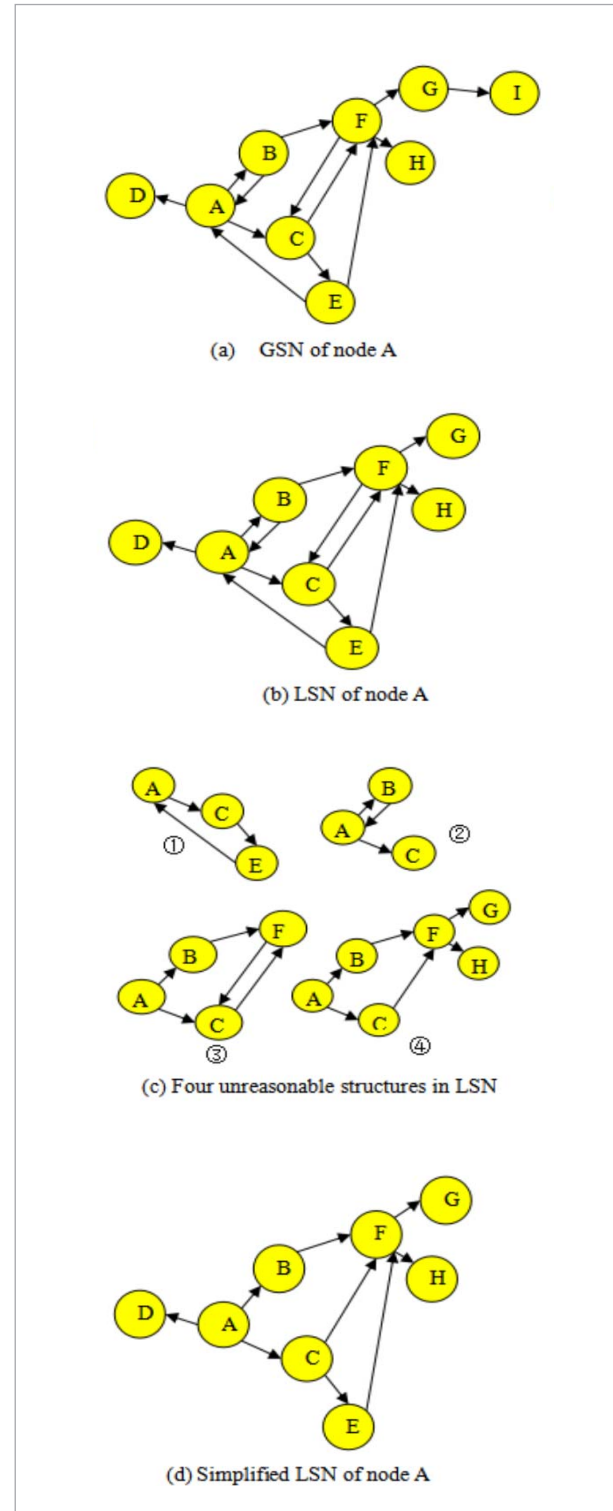
We assume node  $v_A$  as a target user and its GSN is shown in Figure 2(a). To mine the focus of mining  $v_A$ , we first construct the initial LSG of the target user from his/her GSN according to the most effective level (we assume  $l = 3$ ). Thus, we remove node  $v_I$  to construct  $v_A$ 's LSG as shown in Figure 2(b).

In Figure 2(b), the LSG contains four structures. As Figure 2(c), the first three structures will adversely influence computing the similarity between two vertices. It contains a back chain  $A \rightarrow C \rightarrow E \rightarrow A$ . If this chain exists in the digraph, the computer will calculate  $sim(v_A, v_A)$ . However, this process is unreasonable for our objectives, because A should not recommend himself/herself to himself/herself. Hence, we delete edge  $(v_E, v_A)$  before computing the similarity. Figure 2(c) also contains a back edge  $(v_B, v_A)$ , and will likewise misdirect the computer to calculate unreasonable chains, such as  $A \rightarrow B \rightarrow A \rightarrow B$ ,  $A \rightarrow B \rightarrow C$  (B and C are the immediate friends of A, so these users need not be recommended to A). An unreasonable chain similar to edge  $(v_F, v_C)$  also exists in Figure 2(c). Figure 2(c) is reasonable in our algorithm. Thus, we remove all the aforementioned unreasonable edges as above. Then, the simplified LSG is shown in Figure 2(d). We calculate the similarity between the target user and his/her indirect followees according to the calculated user similarity measure. For example, in the HPM, we count the global out/in-degree of the nodes from Figure 2(a). The  $v_A$  is a target user, and the out-degrees of nodes  $v_B, v_C, v_D, v_E, v_F, v_G, v_H$  are 2, 2, 0, 2, 3, 1, and 0, respectively. The in-degrees of nodes  $v_B, v_C, v_D, v_E, v_F, v_G, v_H$  are 1, 2, 1, 1, 3, 1, and 1, respectively. We count the feasible path (FP), number of feasible path  $l = 2$  (NFP2), and the number of feasible path  $l = 3$  (NFP3), indirect neighbor (IN), similarity (by using Equation (5)) of seven vertexes to the target vertexes A from Figure 2(d) in Table 1.

The result shows that vertex F has the highest similarity (0.133) with A, followed by E (0.111), G (0.044), and H(0). The similarities of B, C, and D with A are 1, because they are direct friends of A. We rank users and build the similarity vector as  $\vec{s} = (s_A, s_B, s_C, s_D, s_E, s_F, s_G) = (1,1,1,1,0.133,0.111,0.044)$ . Finally, we get the LSG of the target user as Figure 3.

In this subsection, we propose connection degrees among users. Based on connection degrees  $Cd(i, j)$  and

**Figure 2**  
An Example Constructing LSG for Vertex A





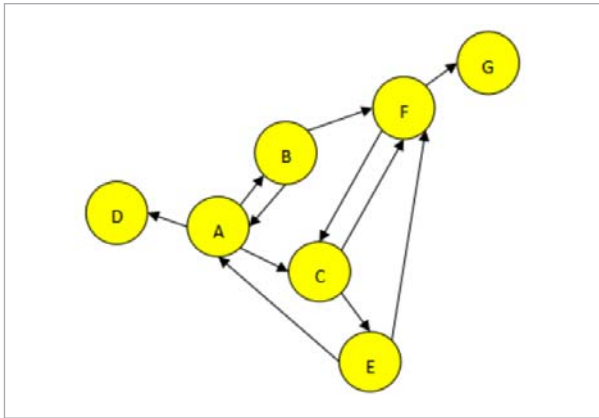
**Table 1**

Chains and Similarities between A and the Nodes of Its LSG

Vertexes	FP	NFP2	NFP3	IN	Sim.
$v_A$					1
$v_B$	$A \rightarrow B$	0	0		1
$v_C$	$A \rightarrow C$	0	0		1
$v_D$	$A \rightarrow D$	0	0		1
$v_E$	$A \rightarrow C \rightarrow E$	1	0	✓	0.111
$v_F$	$A \rightarrow B \rightarrow F$	2	1	✓	0.133
	$A \rightarrow C \rightarrow F$				
	$A \rightarrow C \rightarrow E \rightarrow F$				
$v_G$	$A \rightarrow B \rightarrow F \rightarrow G$	0	2	✓	0.044
	$A \rightarrow C \rightarrow F \rightarrow G$				
$v_H$	$A \rightarrow B \rightarrow F \rightarrow H$	0	2	✓	0
	$A \rightarrow C \rightarrow F \rightarrow H$				

**Figure 3**

LSG of the Target User



active ratio  $AR_j$ , we redefine the relationships among the users by synthesizing them into active score  $AfScore(i, j)$  and propose the algorithm constructing the LSG of a target user.

### 3.3. HPM Algorithm

Armentano [3] proposed a novel approach to find the users' interests by analyzing the content of user posts in microblog platform and developed a followee recommender system by discussing the topology of the microblog network. Stefano [35] classified users of microblogs into different target communities, identified the relevant item sets for diverse communities, and recommended microposts of different users using semantic association rules.

In real-life situation, users always publish posts with a change over time in online social network. In gener-

al, traditional method cannot fully mine a user's focal person. Thus, we first propose the HPM approach to mine the hot personae hidden in microposts.

#### 3.3.1 Extracting the Underlying Personae

Some useful tools, such as Language Technology Platform (LTP) [13], are used to extract the person entities of the micropost. In the LSG, user  $i$  usually reviews some persons. We define these persons as the interested persons of user  $i$ . We can extract all interested persons hiding in the microposts of every user in LSG using these tools.

#### 3.3.2 Mining Hot Personae

In the HPM, we also regard a user as basic unit and denote it as a target user. Two important parameters are considered to mine the hot personae in the microposts.

- The LSG of target user  $j$  is the search space, in which the hot personae are limited. The  $AfScore(i, j)$  is an important weight for each user  $i$ . It reflects the active degree of the user  $i$ . We construct the active score vector  $\vec{s}_j$  of the target user  $j$ . Let  $\vec{s}_j = (s_1, s_2, \dots, s_k, s_j)$  be for the LSG, where  $s_1, s_2, \dots, s_k$  are the  $AfScore(i, j)$ s of users 1, 2, ..., k in the LSG of the target user  $j$ .
- The frequency of instance in which the interested persons occurs in microposts of user  $i$ , is another important parameter. This parameter measures whether the user  $i$  is interested in the person or not. From another point of view, it measures the focal interested person. The number of interested persons of every user in the LSG of the target user  $j$  constructs the user interest matrix  $U$ . Each element  $u_{i, h}$  in  $U$  represents the number  $N_{i, h}$  of the interested person  $h$  of every user  $(i, \text{ or } j)$  in the LSG of the target user  $j$ , and  $k$  is the number of all interested persons in LSG.

Finally, we calculate the user's attention rate vector of the interested persons in LSG as follows:

$$\vec{a} = \vec{s} \times U = (a_1, a_2, \dots, a_k),$$

where  $H_k$  represents the attention rate of the interested person  $k$  in LSG.

The pseudo-code of the HPM approach is shown in Algorithm 3 and the process of the HPM approach for each LSG is described as follows:

**Step 1.** The LSG of target user  $j$  is constructed according to the active score  $AfScore(i, j)$ ;

**Step 2.** The active score vector  $\vec{s}_j = (s_1, s_2, \dots, s_k)$  of the target user  $j$  is constructed;

**Step 3.** The published microposts of each user in the LSG are downloaded;

**Step 4.** The parts of speech for each microposts are tagged, and the persona entities are extracted using LTP;

**Step 5.** The number of the interested persons of every user in the LSG of the target user  $j$  is analyzed;

**Step 6.** The user interest matrix is constructed;

**Algorithm 3:** HPM ( $V, k$ )

```

01 Input
02 LSG; // LSG of V;
03 AfScore(i, j) for GSN;
04 Output
05  $\vec{a}$  // Attention rate vector of hot personae;
06 Begin
07 For each node in V.list;
08   Search  $j$  for  $v \in V$ ;
09   Extract  $\vec{s}_j$  for target user  $j$ ;
10    $\vec{s}_j.add(1)$ ; // add  $s_j$  the active score of target user  $j$ ;
11    $\vec{s}_j \leftarrow \vec{s}_j - \vec{0}$ ; // clean  $s_j$  of non user of LSG;
12   For each user  $i$  in LSG;
13     Load down recent published microposts of user  $i$ ;
14     Tagging the part of speech for posts;
15     Extract interested persons entity from microposts;
16     For each interested person  $h$ 
17       Count  $u_{i,h}$ 
18     End For
19     U.add( $u_{i,h}$ )
20   End For  $i$ 
21    $\vec{a}_j = \vec{s}_j \times U$ ;
22   For  $a_i \in \vec{a}_j$ 
23     If  $i \in \vec{a}_i$  Then
24        $\vec{a}.add(\max(a_i))$ ;
25     Else
26        $\vec{a}.add(a_i)$ ;
27     End If
28   Empty  $U, \vec{a}_j, \vec{s}_j$ ;
29   End For
30 EndFor
31 Rank each  $a_k$  in  $\vec{a}$ ;
32 list Top N of  $\vec{a}$ ;
33 End;
```

In algorithm 3, Steps 12~20 compute each elements  $u_{i,h}$  of  $U$ .  $u_{i,h}$  indicates interest degree which the user  $i$  is interested in the persons occurring in microposts of the user  $i$ . Steps 22~29 compute the user  $i$ 's attention rate vector  $\vec{a}$  of the interested persons in LSG. Step 31 ranks each  $a_k$  in  $\vec{a}$ . Step 32 recommends Top N person as hot personae.

**Table 2**  
Published Entities

Vertexes	Personae entities	Vertexes	Personae entities
$v_A$	Yao Ming, Wang Zhizhi	$v_A$	Yao Ming, Oneal
$v_B$	Ye Li, Yao Ming	$v_B$	Dong Qing
$v_C$	Huang Haibo, Zhang Mo	$v_D$	Liu Qian
$v_E$	Qin Fei	$v_F$	Yao Ming, Liu Xiang
$v_F$	Dong Qing, Liu Qian	$v_F$	Wang Nan
$v_G$	Liu Qian, Li Yundi		

$v_A \sim v_F$ : the real people's Vertexes on the LSN

The similarity vector of the target user is obtained after completing the HPM approach. Then, we obtain the listed users' focal areas from their posts and construct the user-focus matrix to calculate the attention rate vector of the target user. We first download users' posts and extract the personae. Table 2 shows the extracted personae entities (twelve kinds of personae entities in the target user's LSG) in the users' posts. Each row represents a node (A, B, C, D, E, F, or G) and each column represents a persona entity (Yao Ming, Wang Zhizhi, Oneal, Ye Li, Dong Qing, Huang Haibo, Zhang Mo, Liu Qian, Qin Fei, Liu Xiang, Wang Nan, or Li Yundi). The first value 2 represents the attention rate for Yao Ming, that is, two posts mentioned Yao Ming.

$$U_{Table\ 2} = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Finally, we work out the attention rate vector  $\vec{a} = \vec{s} \times U = (3.133, 1, 1, 1, 1.133, 1, 1, 1.177, 0.111, 0.133, 0.133, 0.044)$ . Node A's attention rate vector shows that node A pays most attention to Yao Ming, whose attention rate is 3.133. Dong Qin is not mentioned in A's posts. However, his attention rate is 1.133, because A's friends B and F published posts about Dong Qin. Thus, Dong Qin may be A's focus.

In this subsection, we extract the person entities from Microblog posts of LSG, calculate the user's attention rate to mine hot personae, and propose an algorithm mining hot personae.

## 4. Experimental Evaluation

In this section, we introduce our experimental settings, which include the data sets, parameter settings, program flows and evaluation indices. Then, we discuss the results of evaluations.

### 4.1. Experimental Settings

We obtain 4 data sets (i.e., DS1, DS2, DS3, and DS4) from Tencent Microblogs using Tencent's API. the number of all users (NAU), and the number of relevant users (NRU) are shown in Table 3.

In our evaluation, we define the following evaluation

**Table 3**

Four Data sets

	NAU	NRU	NAU	NRU
DS1	3458	283	DS3	7166
DS2	7586	1065	DS4	3327
				651

metrics to measure the performances of our proposed approach and other four approaches:

- Precision (P): The ratio of the number of relevant users to the number of users in the top-k recommended users;
- Recall (R): The ratio of the number of relevant users in the top-k recommended users to the total number of relevant users in the data set;
- F-measure (F): This value synthesizes  $R$  and  $P$  and reflects the comprehensive performance of the approaches.  $F = \frac{2 * P * R}{P + R}$ .

In Equation (7) and Algorithms 1 and 2, the parameter  $l$  is very important to construct local social network from the GSN. When parameter  $l$  takes 2 and 3, we retrieve the precisions (Table 4) and recalls (Table 5) under Top10, Top20, Top30, Top40, Top50 of the user lists that our HPM algorithm recommend from DS1, DS2, DS3, DS4.

We found that the smaller the  $l$  is, the higher the precision of HPM algorithm will be. Moreover, the recall

of HPM algorithm will become lower. A higher value of  $l$  will result in lower precision but the more recall of HPM algorithm. Tables 4 and 5 show that the precision and recall of the HPM algorithm are relatively stable when  $l = 2$  and  $l = 3$ . This phenomenon is due to the lower numbers of users of link path length  $l = 3$  from the different users to the same user than link path length  $l = 2$  in social networks. Evident changes take place in the  $l = 1$  and  $l = 4$ . The numbers of users of link path length  $l = 1$  from the different users to the same user are obviously more than link path length  $l = 2$ . The numbers of users of link path length  $l = 4$  from the different users to the same user are obviously less than link path length  $l = 3$ . Thus, we consider  $l = 2$  or  $l = 3$  in our experiments.

**Table 4**

Precisions of HPM When  $l = 2$  and  $l = 3$

DS	$l$	Top 10	Top 20	Top 30	Top 40	Top 50
DS1	$l = 2$	50.00	45.00	46.67	47.50	46.00
	$l = 3$	50.00	45.00	46.67	50.00	48.00
DS2	$l = 2$	30.00	35.00	26.67	27.50	26.00
	$l = 3$	40.00	35.00	30.00	32.50	28.00
DS3	$l = 2$	30.00	40.00	40.00	40.00	40.00
	$l = 3$	40.00	40.00	43.33	42.50	42.00
DS4	$l = 2$	40.00	40.00	43.33	50.00	54.00
	$l = 3$	40.00	40.00	43.33	52.50	56.00

### 4.2. Result Analysis

In this subsection, we compare our proposed HPM approach with the classical AA [1], FOFA [6], LHNI [18], and FL [30] approaches for based on the precision, recall, and F-measure using four data sets.

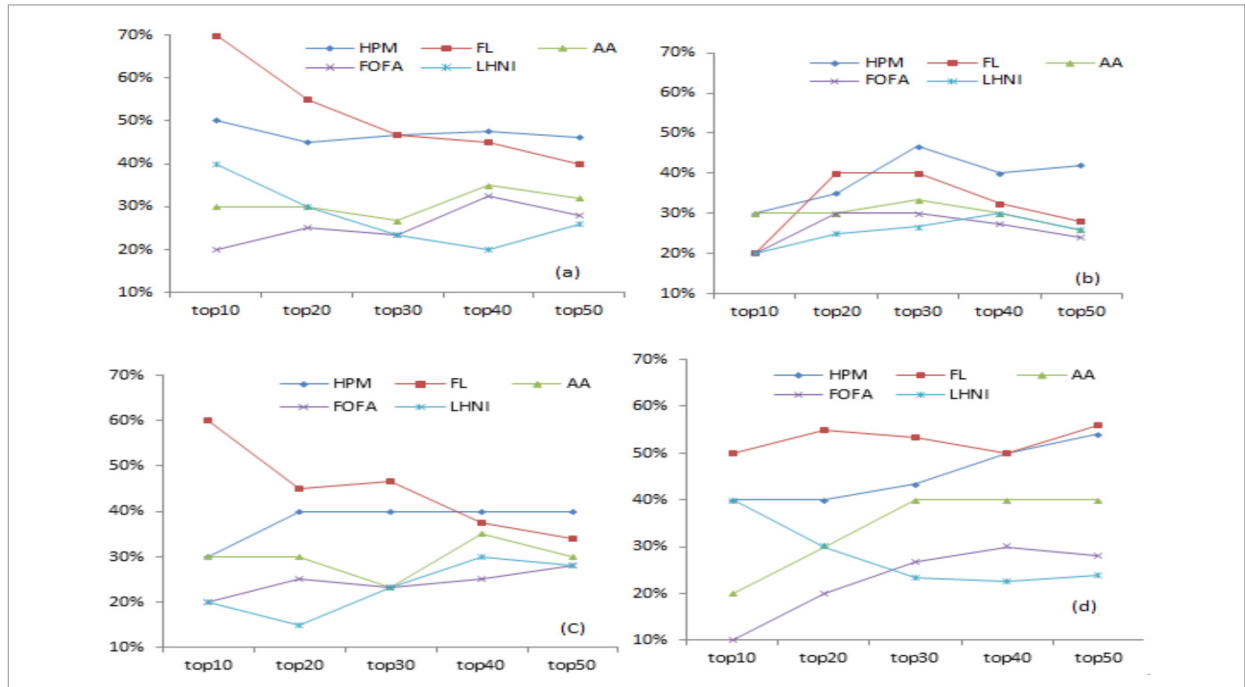
Figure 4 demonstrates a comparison of the precisions of different approaches (HPM, FL, AA, FOFA, LHNI) on Top N values of the data sets. The general tendencies of the precision indicate that our proposed HPM performs better than the other approaches. Thus, HPM can efficiently mine the hot personae. In Figures 4(a)~(d), the precisions of our proposed HPM and FL are prominently higher than AA, FOFA, and LHNI. However, the precision of our proposed HPM is lower than that of FL from the Top10 to the Top30 but higher than that of FL from the Top30 to the Top50 in Figure 4(a). This result proves that HPM is suitable for mining the deep hot personae. The precision of our proposed HPM is lower than that of FL from the Top10 to the Top25 in Figure 4(b). The precision of our proposed HPM is lower than that of FL from the

Top10 to the Top40 but higher than that of FL from the Top40 to the Top50 in Figures 4(c) and (d).

Figure 5 demonstrates a comparison of the recall of different approaches (HPM, FL, AA, FOFA, and LHN I).

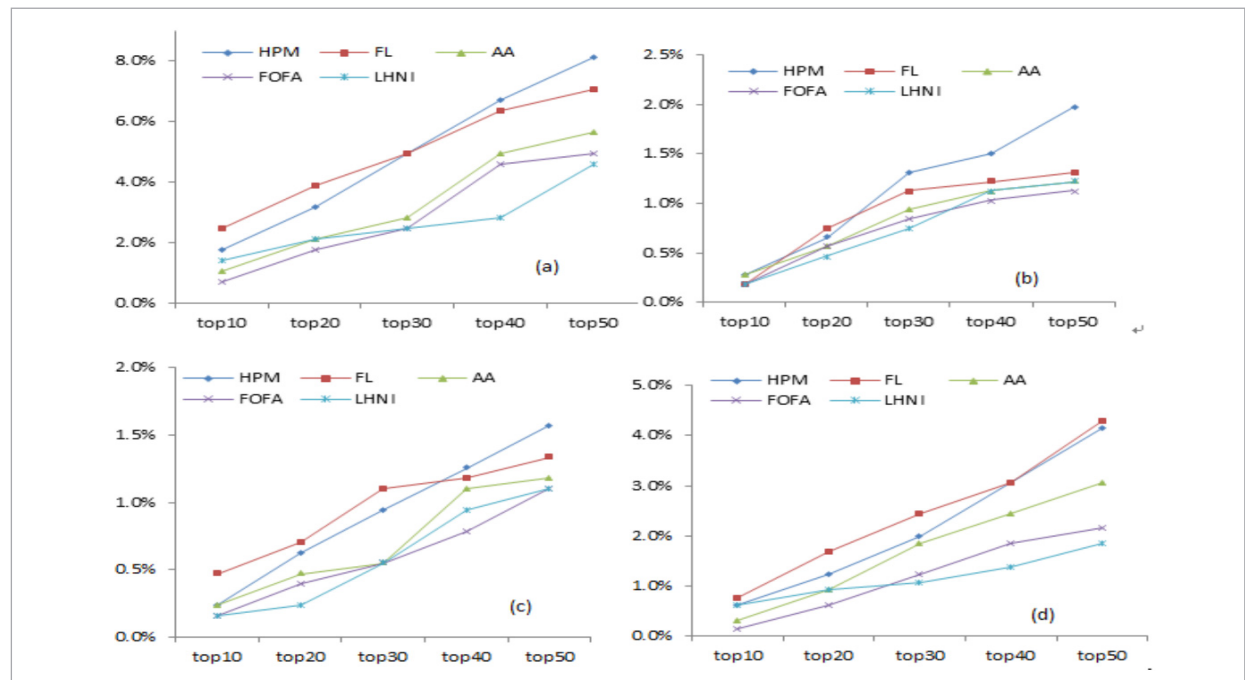
**Figure 4**

A Comparison of Different Approaches in Precision



**Figure 5**

A Comparison of Different Approaches in Recall

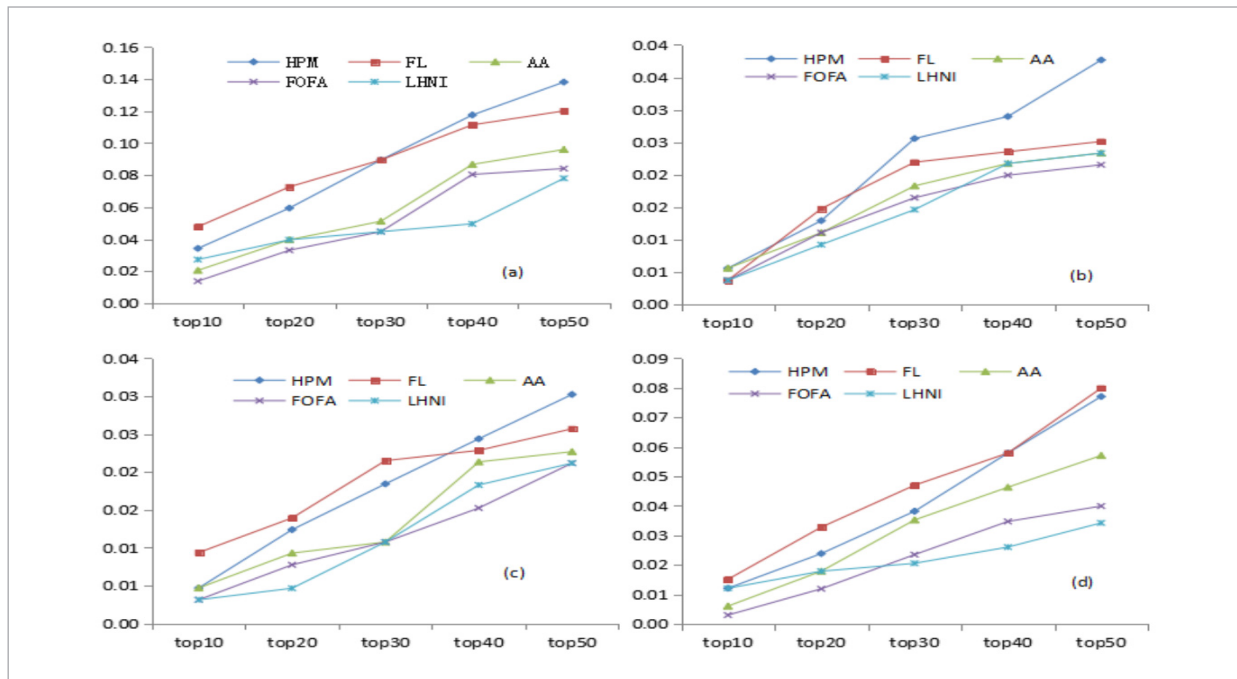


LHNI) on Top N values for the four data sets. The general tendencies of the recall indicate that our proposed HPM outperforms AA, FOFA, and LHNI. Our proposed HPM falls behind FL in some aspects. However, the result proves that HPM still is an efficient approach for mining the hot personae. In Figures 5(a)~(d), the recall of our proposed HPM and FL are prominently lower than those of AA, FOFA, and LHNI. However, the recall of our proposed HPM is

lower than that of FL from Top10 to Top30, and the recall of our proposed HPM is higher than that of FL from Top 30 to Top50 in Figures 5(a)~(b). However, recall of our proposed HPM is lower than that of FL from Top10 to Top40 but higher than that of FL from Top40 to Top50 in Figures 5(c)~(d).

Figure 6 demonstrates a comparison of the F-Measure of the different approaches (HPM, FL, AA, FOFA, and LHNI) on different Top N values for four

**Figure 6**  
A Comparison of Different Approaches in F-Measure

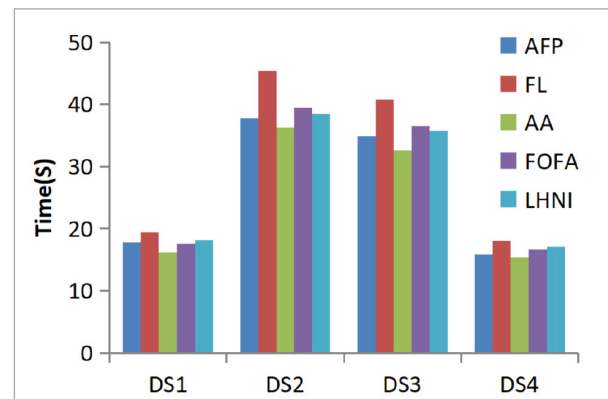


data sets. The general tendencies of the F-measures for HPM, FL, AA, FOFA, and LHNI are the same as Figures 4 and 5.

Figure 7 provides the time consumed by the five approaches in mining the hot personae. The FL algorithm consumes the most time, and the AA algorithm consumes the least time on DS1 DS2, DS3, and DS4. Our proposed HPM algorithm consumes less time than FL, FOFA, and LHNI algorithms on DS1 DS2, DS3, and DS4. The time efficiency of our proposed HPM algorithm is close to that of AA algorithm.

Table 6 lists the average precision (AvgPre), recall (AvgRev), F-measures (Fms), and total number (Nump) mining hot personae. HPM yields the highest

**Figure 7**  
Time Efficiency of Five Algorithms



Nump (30.4) among the Nump of FL, TFIDE, FOFA, and LHNI. HPM also shows the highest average precisions. The average values for recall and F-measures(Fms) of HPM are 8.72% and 12.9%, respectively. These values are also the highest among the five approaches.

## 5. Conclusion and Future Work

- In this paper, we put forward a novel approach to discover hot personae, who are reviewed hotly by users from the microblog posts. The proposed approach includes the following parts:
- We define the active ratio by considering the followees and followers of every nodes to measure the importance of the personae corresponding to nodes.
- We define connection degrees from nodes to the other nodes by considering the active ratios of every node and the path length. We denote nodes of the max active ratios to the center nodes and construct the LSG of the nodes. We select the nodes where  $Cd$  is more than the given threshold as nodes in LSG.
- We retrieve all the microblog posts of the LSG and extract all personae hidden the microblog posts. Then, we construct the interest matrices and compute the  $AfScore$ . The top  $k$  personae are ranked as the hot personae.

Our experiments prove that the HPM algorithm is efficient for mining the hot personae.

- We take the precisions and recalls of HPM in different link path length  $l$ . On DS1, DS2, DS3, and DS4, the parameter  $l$  of our proposed HPM algorithm takes 2 or 3.
- We compare the precisions, recalls and F-Measures of HPM with FL, AA, FOFA, and LHNI approaches on Top N values of the data sets. The precisions and recalls of our proposed HPM are lower than that of other four approaches from the Top10 to the Top40

but higher than that of other four approaches from the Top40 to the Top50.

- Moreover, the HPM reduces the time complexity compared with FL, AA, FOFA, and LHNI approaches by dividing the whole social networks into the LSG. Our proposed HPM algorithm consumes the less time than FL, FOFA, and LHNI algorithms on DS1 DS2, DS3, and DS4. The time efficiency of our proposed HPM algorithm is close to the least time of AA algorithm.
- HPM yields the highest Nump (30.4) among the Nump of FL, AA, FOFA, and LHNI. HPM also shows the highest average precisions (24.6%). The average values for recall and F-measures(Fms) of HPM are 8.72% and 12.9%, respectively. These values are also the highest among the five approaches.

In the future, we will perform a more comprehensive study on the hot persona mining, which will mainly include the following aspects:

- Although, we have performed numerous experiments based on the four data sets, and some results prove that our proposed algorithm outperforms FL, AA, FOFA, and LHNI. The data sizes are not sufficient. We need to retrieve more data to demonstrate the performances of our proposed approaches.
- In the HPM algorithm, we still consider the semantic meanings of microposts to discover the hot personae.
- We will develop applications using the HPM algorithm. We will discuss whether the HPM algorithm can extend other social networks.

## Acknowledgments

This work is one of the projects supported by National Nature Science Foundation (Grant Nos. 61472329, 61532009 and 61872298), and supported by Sichuan Science and Technology Program (Grant No. 2018GZ0096). We especially thank Prof. Xiaofei He of Zhejiang University for his helpful comments and suggestions.

## References

1. Adamic, L., Adar, E. Friends and Neighbors on the Web. *Social Networks*, 2003, 25(3), 211-230. [https://doi.org/10.1016/S0378-8733\(03\)00009-1](https://doi.org/10.1016/S0378-8733(03)00009-1)
2. Allen, S. M., Chorley, M. J., Colomboa, G. B., Jahob, E., Karaliopoulosb, M., Stavrakakisb, I., Whitaker, R. M. Exploiting User Interest Similarity and Social Links

- for Micro-Blog Forwarding in Mobile Opportunistic Networks. *Pervasive and Mobile Computing*, 2014, 11, 106-131. <https://doi.org/10.1016/j.pmcj.2011.12.003>
3. Armentano, M. G., Godoy, D., Amandi, A. A. Followee Recommendation Based on Text Analysis of Micro-Blogging Activity. *Information Systems*, 2013, 38(8), 1116-1127. <https://doi.org/10.1016/j.is.2013.05.009>
  4. Behera, R. K., Rath, S. K., Misra, S., Damasevicius, R. Large Scale Community Detection Using a Small World Model. *Applied Sciences*, 2017, 7(11). <https://doi.org/10.3390/app7111173>
  5. Chen, H. H., Cui, X. L., Jin, H. Top-k Followee Recommendation over Microblogging Systems by Exploiting Diverse Information Sources. *Future Generation Computer Systems*, 2016, 55, 534-543. <https://doi.org/10.1016/j.future.2014.05.002>
  6. Chen, J., Werner, G., Casey, D., Michael, M., Ido, G. Make New Friends, but Keep the Old: Recommending People on Social Networking Sites. *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, ACM, April 4-9, 2009, 201-210. <https://doi.org/10.1145/1518701>
  7. Chen, K., Zhou, Y., Zha, H. Y., He, J. H., Shen, P., Yang, X. K. Cost-Effective Node Monitoring for Online Hot Event Detection in Sina Weibo Microblogging. *Proceedings of 22nd International Conference on World Wide Web*, May 13-17, 2013, 107-108. <https://doi.org/10.1145/2487788.2487837>
  8. Cleveland, W. S., Devlin, S. J. Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *Journal of the American Statistical Association*, 1988, 83(403), 596-610. <https://doi.org/10.1080/01621459.1988.10478639>
  9. Dao, T. H., Jeong, S. R., Ahn, H. A Novel Recommendation Model of Location-Based Advertising: Context-Aware Collaborative Filtering Using GA Approach. *Expert Systems with Applications*, 2012, 39(3), 3731-3739. <https://doi.org/10.1016/j.eswa.2011.09.070>
  10. Dong, Y., Tang, J., Wu, S., Tian, J. L., Chawla, N. V., Rao, J. H., Cao, H. H. Link Prediction and Recommendation Across Heterogeneous Social Networks. *Proceedings of 2012 IEEE 12th International Conference on Data Mining*, December 10-13, 2012, 181-190. <https://doi.org/10.1109/ICDM.2012.140>
  11. Fahd, K., Habib, N. E., Omar, E. B. Microblog Semantic Context Retrieval System Based on Linked Open Data and Graph-Based Theory. *Expert Systems with Applications*, 2016, 53, 138-148. <https://doi.org/10.1016/j.eswa.2016.01.020>
  12. Fan, M., Zhou, Q., Zheng, T. F. Mining the Personal Interests of Microbloggers via Exploiting Wikipedia Knowledge. *Proceedings of Computational Linguistics and Intelligent Text, CICLing 2014*, April 6-12, 2014, 84(04), 188-200. [https://doi.org/10.1007/978-3-642-54903-8\\_16](https://doi.org/10.1007/978-3-642-54903-8_16)
  13. Harbin Institute of Technology, Iflytek CO., LTD. Available at <http://www.ltp.cloud.com>, 2019.
  14. Hua, W., Huynh, D. T., Hosseini, S., Lu, J. Information Extraction from Microblogs: A Survey. *International Journal of Software and Informatics*, 2012, 6(4), 495-522.
  15. Ji, Z. Y., Pi, H. Y., Wei, W., Xiong, B., Marcin, W., Robertas, D. Recommendation Based on Review Texts and Social Communities: A Hybrid Model. *IEEE Access*, 2019, 7, 40416-40427. <https://doi.org/10.1109/ACCESS.2019.2897586>
  16. Jung, J. J. Online Named Entity Recognition Method for Microtexts in Social Networking Services: A Case Study of Twitter. *Expert Systems with Applications*, 2012, 39, 8066-8070. <https://doi.org/10.1016/j.eswa.2012.01.136>
  17. Kim, Y., Cho, N. W. Research Trends in Social Network Analysis Using Topic Modeling and Network Analysis. *ICIC Express Letters*, 2018, 12(1), 71-78.
  18. Leicht, E. A., Holme, P., Newman, M. E. J. Vertex Similarity in Networks. *Physical Review E*, 2006, 73(2), 02612001-02612010. <https://doi.org/10.1103/PhysRevE.73.026120>
  19. Leila, W., Paulo, Q., Oliveira, P. J. Measuring Node Importance: A Multi-Criteria Approach. *Proceedings of IADIS International Conference WWW/Internet, ICWI 2011*, November 5-8, 2011, 415-420.
  20. Li, F., Du, Y. J., Zhao, H., Feng, Z. G. Two-Phase Strategy of Chinese Named Entity Recognition in Micro-Blog. *Journal of Computational Information Systems*, 2014, 10(19), 8421-8428. DOI: 10.12733/jcis11989.
  21. Liu, J., Du, Y. J., Li, Q., Fu, C. L. Social Community Evolution by Combining Gravitational Relationship with Community Structure. *Intelligent Data Analysis*, 2018, 22(5), 1143-1161. <https://doi.org/10.3233/IDA-173561>
  22. Liu, T., Che, W. X., Li, Z. H. Language Technology Platform. *Journal of Chinese Information Processing*, 2011, 25(6), 53-62.
  23. Liu, Z. Y., Chen, X. X., Sun, M. S. Mining the Interests of Chinese Microbloggers via Keyword Extraction. *Frontiers of Computer Science*, 2012, 1(6), 76-87. DOI: 10.1007/s11704-011-1174-8.
  24. Lu, J., Wan, W. G. Identification of Key Nodes in Microblog Networks. *ETRI Journal*, 2016, 38(01), 52-61. <https://doi.org/10.4218/etrij.16.0115.0732>

25. Lü L. Y., Zhou, T. Link Prediction in Complex Networks: A Survey, *Physica A: Statistical Mechanics and Its Applications*, 2011, 390(6), 1150-1170. <https://doi.org/10.1016/j.physa.2010.11.027>
26. Ma, J. W., Hao, X., Chen, H. H. Friendship Prediction in Recommender System. *Journal of National University of Defense Technology*, 2013, 35(1), 163-168.
27. Martincic-Ipsic, S., Močibob, E., Perc, M. Link Prediction on Twitter. *PLoS ONE*, 2017, 12(7), e0181079. <https://doi.org/10.1371/journal.pone.0181079>
28. Nettleton, D. F. Data Mining of Social Networks Represented as Graphs. *Computer Science Review*, 2013, 7, 1-34. <https://doi.org/10.1016/j.cosrev.2012.12.001>
29. Nocoletti, M., Schiaffino, S., Godoy, D. Mining Interests for User Profiling in Electronic Conversations. *Expert Systems with Applications*, 2013, 40, 638-645. <https://doi.org/10.1016/j.eswa.2012.07.075>
30. Papadimitriou, A., Symeonidis, P. Manolopoulos, Y. Fast and Accurate Link Prediction in Social Networking System. *Journal of system and software*, 2012, 85(9), 2119-2132. <https://doi.org/10.1016/j.jss.2012.04.019>
31. Paul, M. Technically Speaking: All a-Twitter, *IEEE Spectrum*, 2007, 44(10), 84-84. <https://doi.org/10.1109/MSPEC.2007.4337670>
32. Peng, B., Du, Y. J., Fan, Y. Q. An Active-Friend Recommendation Algorithm in Microblog. *Journal of Computational Information Systems*, 2015, 11(5), 1657-1664. DOI: 10.12733/jcis13280.
33. Peng, B., Du, Y. J., Fan, Y. Q. Extracting Implicit and Explicit Focusing Personae of User in Microblog. *Journal of Computational Information Systems*, 2015, 11(9), 3441-3448. DOI: 10.12733/jcis14388.
34. Quan, Y., Gao, C., Thalmann, N. M. Enhancing Naive Bayes with Methods for Short Text Classification. *Proceedings of 21st Annual Conference on World Wide Web Companion*, April 16-20, 2012, 645-646. <https://doi.org/10.1145/2187980>
35. Stefano, F., Giovanni, S., Paola, V. A Semantic Recommender for Micro-Blog Users. *Proceedings of 2015 IEEE International Congress on Big Data*, August 17, 2015, 55-62. DOI: 10.1109/BigData Congress.2015.18.
36. Symeonidis, P., Iakovidou, N., Mantas, N., Manolopoulos, Y. From Biological to Social Networks: Link Prediction Based on Multi-Way Spectral Clustering. *Data Knowledge Engineering*, 2013, 87, 226-242. <https://doi.org/10.1016/j.datak.2013.05.008>
37. Tang, J. Y., Liu, Z. Y., Sun, M. S., Liu, J. H. Portraying User Life Status from Microblogging Posts. *Tsinghua Science and Technology*, 2013, 18(2), 182-195. <https://doi.org/10.1109/TST.2013.6509101>
38. Tong, H., Faloutsos, C., Pan, J. Y. Random Walk with Restart: Fast Solutions and Applications. *Knowledge and Information Systems*, 2008, 14(3), 327-346. <https://doi.org/10.1007/s10115-007-0094-2>
39. Wei, W., Cong, G., Miao, C. Y., Zhu, F. D. Learning to Find Topic Experts in Twitter via Different Relations. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(7), 1764-1778. <https://doi.org/10.1109/TKDE.2016.2539166>
40. Xu, L., Jiang, C. X., Ren, Y., Chen, H. H. Microblog Dimensionality Reduction-A Deep Learning Approach. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(7), 1779-1789. <https://doi.org/10.1109/TKDE.2016.2540639>
41. Yao, W. B., Yang, Y. W., Wang, D. B. UR Rank: Micro-Blog User Influence Ranking Algorithm Based on User Relationship. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 2018, 252, 394-404. [https://doi.org/10.1007/978-3-030-00916-8\\_37](https://doi.org/10.1007/978-3-030-00916-8_37)
42. Yin, D., Hong, L. J., Davison, B. D. Structural Link Analysis and Prediction in Microblogs. *Proceedings of 20th ACM International Conference on Information and Knowledge Management*, ACM, CIKM'11, October 24-28, 2011, 1163-1168. <https://doi.org/10.1145/2063576.2063743>
43. Yu, H. K., Zhang, H. P., Liu, Q., Lv, X. Q., Shi, S. C. Cascaded Hidden Markov Model. *Journal on Communications*, 2006, 27(2), 87-94. <https://doi.org/10.1360/jos172601>
44. Yu, S. J. The Dynamic Competitive Recommendation Algorithm in Social Network Services. *Information Sciences*, 2012, 187, 1-14. <https://doi.org/10.1016/j.ins.2011.10.020>
45. Zhao, J. Q., Gui, X. L., Tian, F. A New Method of Identifying Influential Users in the Micro-Blog Networks. *IEEE Access*, 2017, 5, 3008-3015. <https://doi.org/10.1109/ACCESS.2017.2672680>