Improving Content Recommendation in Social Streams via Interest Model

Junjie Zhang and Yongmei Lei

Abstract The current microblog recommendation approaches mainly consider users' interests. But because user's interests are changing dynamically and they have low activity, it's hard to build user interest model. In this paper, we propose a new approach to recommend information based on multiaspect similarities of interest and new dynamic strategy for defining long-term and short-term interests according to user's interest changing. Recommended information is ranked by two factors: the similarity between user's interest and information, tie-strength of user interest. We implemented three recommendation engines based on Sina Microblog and deployed them online to gather feedback from real users. Experimental results show that this method can recommend interesting information to users and improve the precision and stability of personalized information recommendation by 30 %.

Keywords Recommender system · Naive bayes · Interest model · Microblog

1 Introduction

Information overload has appeared in many places, such as websites, e-commerce and so on, but social network has solvedthe problem of information overload for us to some extent. We read information, which send by our friends, so we avoid much information that we are not interested. But when we follow more people, we will get more information, we should spend more time to choose what we are interested. In the end, information overload appears in social network.

J. Zhang (✉) · Y. Lei
Computer Engineering and Science, Shanghai University, Shanghai, China
e-mail: jadsin@shu.edu.cn

Y. Lei
e-mail: lei@shu.edu.cn

R. Lee (ed.), *Computer and Information Science*, Studies in Computational Intelligence 566, DOI 10.1007/978-3-319-10509-3_5

Recommender can solve information overload problem effectively. The most famous method is collaborative filtering (CF), which can recommend valuable information according users' preferences. This approach can be divided into different types according to different directions, such as collaborative filtering based on items [9], collaborative filtering based on users, but they all require users to rate items and record their purchase history.

However social network is different, they have no items to rate, although users can rate every story, but this will lose much information contained in these stories. For recommending micro-blog, researchers are more like to use other information such as textual content [1, 17] and LDA topic model [10] etc.

There are many recommendation researches based on micro-blog, Yu et al. [19] transmit relationships into graph, then recommend friends based on mixed graph. Weng et al. [18] researched how to find topic-sensitive influential twitterers; Chen J regarded three factors for recommending conversations in online social streams [4]. In another paper [5], Chen J introduced URL recommendation on Twitter, they used three properties: recency of content, explicit interaction among users and user-generated content.

Recent researches research more about recommending friends, conversations and so on, they referred to comments, tags or replays. However, there are few researches concerning content recommendation, and not all people in social network are interested in all the information. To save time for user to get interesting information, social networks provide effective methods, Facebook used EdgeRank algorithm [8], particularly favoring recent and long conversations related to close friends, to recommend social stream. Instead, Twitter adopts simple rules to filter streams. Sina micro-blog has a function called intelligent ranking, which can recommend information, but if users don't use the system usually, its performance begins to decline. However, we don't know how they realized.

It's hard to find interesting information for user because of four difficulties:

- It's difficult to build users' interests. No matter online or offline, user's interests are changing dynamically. Although content can be considered as containing user interest, but it is valuable within a short time since being published.
- Low activity. Unlike recommendation system, people rate items, and then system figures out people's interests according these scores. In social networks, people are more like to read information rather than write or rate, so it's difficult for us to get their explicit information and interests.
- Limit of content's words. In social network, every status is limited within 140 words. To express within limit words, users minimize the usage of repeated words. It's hard to analysis contents using existing models.
- Large amount of data but sparse. Users provide few explicit information in social network, so we use large amount of implicit information, such as information contents, time, tags or followees' interests. However, although there are many followees, most of them can't provide effective interests.

To solve these problems when recommending content, we propose an improved algorithm of building user's interests and measuring the similarity between user and content, and then recommend the most similarity information. We deploy recommendation system using our algorithm online to gather feedback from real users. We asked users to rate for each recommending information produced by different algorithms. The result scores can help us to compare the performance of different algorithms.

The rest of the paper is structured as follows. First, we discuss how existing research relates to our work. We then propose an improved algorithm and detail our design of algorithm and deploy recommendation system for studying, and then we detail our work with results. We conclude with comparison of our recommendation to others, such as Sina Intelligent Ranking.

2 Related Work

As popular of social network, more and more people use it to get information. But we'll find that we are not interested in every information that posted by our friends, and what's more, because people's preferences are quite different. Paek et al. [12] found that many posts were considered important to one user but worthless to another user, that is personal preferences is important for content recommendation. In social networks, there is little explicit information to build user's interests, we can't rate items to indicate our preferences. But statuses contents that posted by us can represent. We can extract key words that appear many time in our posted contents to indicate our interests.

There are many recommendations based on textual content, such as websites [13] and books [11]. For example, to recommend websites, Pazzani et al. [13] created bag-of-word profiles for individuals from their activities, and chose websites that are most relevant to the profile as recommendations. Because activities of an individual are often insufficient for creating useful profiles, Balabanovic et al. created profile from a group of related individuals [1].

Since the fast update rate of micro-blog and users' interests are changing dynamically, we divide users' interests into long-term interests and short-term interests. To improve the precision of recommendation, we consider three factors to build users' interests:

- Tags: People can use tags to represent their interests. We can also find users that has common interests through tags.
- Contents: When users send a status, they may want to transmit information or express their status, but these all contains their interests. We use the contents of the latest one month to build user's short-term interest.
- Followees: In social networks, most people use it to get information, they rarely write news, those types of people have low activities. We follow someone to get useful information from them, we are interested in what followees post and have some common interests with them, so when constructing interests, we consider

using their followees' interests to extend their interests, especially when users have low activities. Prior work [16] had proved the effectiveness of combining text content from a group to capture the interest of single user.

As mentioned earlier, user's contents are interesting within a period of time since being published. Boyd, D mainly researched the conversational usage of retweets in Twitter, he thought the time that tweets are posted was important [3]. When considering contents to build interests, we also consider the influence of time.

In social network, the importance of individuals is different. For example, my friend David, he always read information from sina micro-blog, but he posts few, so he is at a normal position , once he post a status, we may think his information is not very valuable. And my friend Ange, she reads information as well as writes always, we always get information from her posts, so we think her news or interests are more important than David to us. So we divided followees, when using their interests to construct user's interests, we attach different importance to different followees.

Tie-strength is a characterization of social relationships between people [7]. Gilbert et al. [6] suggested that tie-strength may be a useful factor for filtering messages in Facebook news feeds and measured tie-strength with many dimensions. Chen et al. [4] was inspired by Gilbert, they used three dimensions to recommend conversions: existence of direct communications, frequency of such direct communications, tie-strength between the two and their mutual friends and Chen J et al. Measured tie-strength between users according to the frequency of communications among users and tie-strength between the two and their mutual friends.

People follow someone because of their common interests, so we follow common friends indirectly indicating that we have common interests. Inspired by Chen, when measuring tie-strength between users, besides considering the similarity between their owns' interests, we also consider their common friends' tie-strength of interests.

When we want to recommend information to users, we calculate the similarity between user's interests and content and tie-strength between users based on interests, and then recommend high scores information.

3 Multiaspect Similarity of Interest

To find interesting information for users in social streams, we propose a new method to recommend information based on multiaspect similarities of interest, which considers the similarity between user's interests and information and tie-strength between users' interest. Due to dynamical changing of interest, we use their long-term and short-term interests to build users' interest model, and we don't merely use total number of common friends to measure tie-strength, we also take their common friends' interests into account, which will improve the precision in social network based on interests. We recommend the most similarity information to users according their interests. Next we will introduce our method through three parts.

3.1 Interest Model

Due to the dynamical changing of user's interests, we divided interests into long-term and short-term interests, which can build user interests more precise.

Users' short-term interests show change of his recent statuses. In our research, we followed the approach in Pazzani et al. [13], we build a bag-of-words profile for each user to represent his short-term interests. Unlike in Pazzani et al., where the profile consists of words from web pages that the user has rated explicitly, we build the profiles from users' recent one month micro-blog contents with TF-IDF model [15].

However, since each status contains less than 140 words, the constructed bag-of-words had lower weight, so we use TweeTopic technique [2] to enrich the content of each status. We feed each micro-blog to Google CSE [14] search engine, and extract the returned documents to get key words to represent this information's vector.

Users are more interested in recent news, and the value of information begin to decay with time passing, so to predict users' interests better, we modified TF-IDF model, and added time penalty function as (1)

$$f(u) = \sum_{n=1}^{N} \sum_{x} (TF_x{}^* IDF_x{}^* T_n) \tag{1}$$

where N is total number of user's micro-blogs, x is words in document n. $\mathrm{TF_x}$ is the frequency of word x in micro-blog n; IDF_x is log(total number of micro-blogs / total number of micro-blogs that contain word x). $\mathrm{T_n}$ is time penalty function, there are many functions, we use one of them as (2)

$$T = \frac{1}{1 + \alpha|T_0 - T_i|} \tag{2}$$

where T0 represents current time, Ti represents time of micro-blog i that be posted, α is weight. When α is assigned to different value, the decay speed is different. Experiment shows that when α is assigned to 0.5, the decay speed is better.

Unlike short-term interests, users' long-term interests don't change usually, but it can increase. User posts a status, which is valuable within a short time, and when time is passing, the value of this status is decaying, so old statuses only show that users had ever been interested in them, we can't only use contents to build use's interests.

In our research, when building long-term interests, we consider three factors: micro-blog's contents, tags and followees' interests. For most people read information rather than write, using contents and tags merely to build interests maybe not precise. We are inspired by Chen [5], they researched URLs recommendation on Twitter, when building URLs candidate, they considered URLs posted by followees and followees of followees. So we use user's followees' interests to enrich their interests.

However, we have analyzed the recent one month contents for constructing short-term interests. So to build long-term interests, we don't need to analyze these contents any more. We had built all short-term interests every month, then we can reuse these, but we should add time factor, but this time we use month as unit.

In social relationship, we can't treat every followees as equal, Some people, who have unique views and post many valued information, are extremely important. In our research, we divide followees into two types: Information Source User and Normal User. Information Source Users always provide unique and valued information or unique Opinions on something, most of us like to read news from them, hoping to extend our interests according to their own interests, then we can get more and more interesting and valuable information. But Normal Users are not as import as the other type. They mostly like to read and don't usually express their opinions. So we will attach more importance to Information Source Users than Normal Users.

We use Naive Bayes to divide them, the method includes four characters: number of followees, number of fans, number of bi-followers and number of micro-blogs, the classifier as (3)

$$P(C_i|W) = \frac{P(C_i)\prod_{n=1}^{4} P(W_i|C_i)}{P(W)} \tag{3}$$

where W is consist by four characters, C is the two types.

We build user's followees' interests as follows: for each followee F, if he has been built interest vector, then we use it directly. Otherwise, we use F 's all contents that he posted, we then use the modified TF-IDF models to analysis these contents, but we don't feed micro-blogs to search engine for improving efficiency, we get an interest vector as V_F, rank the vector by decreasing order of words' weight in V_F, then we remove these words that have low weight, and choose the top 10 % of words to extend user's long-term interests.

During this operation, we gather many words from their followees, and the weight of every word may be bigger. When weight of words becomes bigger, that shows more than one followee are interested in these words, and user is more interested in them. So after gathering all followees' interest vector into one, we set a threshold, only can the weight is larger than the threshold be kept. This will decrease complexity of future computing.

We build user's long-term interests with tags, contents and followees' interests with weight. And we add long-term and short-term interests to construct user's interest vector, because users are more eager to get information, what he pays attention to recently, and we give higher weight to short-term interests than long-term interests. This will improve recommendation's breadth without losing accuracy of recommendation.

3.2 Tie-Strength

Micro-blog's friendship is based on common interests, when users follow the same people, they are more likely to have common interests. If we measure tie-strength by total number of common friends, that lose much information behind relationship, such as whether we really both have common interests with our common friends, whether we are more familiar to user A than user B, besides, that can't make a distinction between information source users and normal users as well.

So in our research, when measuring tie-strength between users, we don't use total number of common friends, we consider the tie-strength of their common friends' interests with them and the similarity between users own interests. To distinguish between information source users and normal users, we give higher weight to information source users. The detail of this approach as follow, we assume to calculate tie-strength between A and B in Fig. 1.

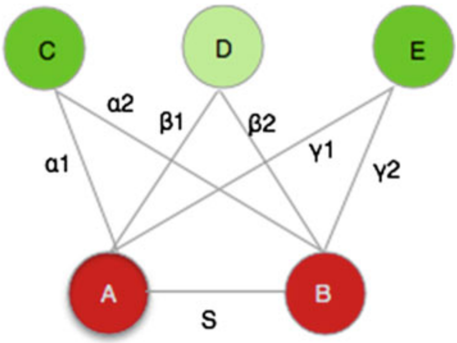
In Fig. 1, C, D and E are common friends of user A and user B, C and E are information source users, they have higher weight as a in (4), D is normal user, he has low weight. $\alpha_1$is the tie-strength between A and C, others define as same. We suppose A and C's common interests are $<V_1, V_2, \ldots, V_n>$, A's interest vector mapping in common interests with weight is $<W_1, W_2, \ldots, W_n>$, and is called V_a, and C is $<W'_1, W'_2, \ldots, W'_n>$, called V_c, then we use cosine similarity to calculate α_1

$$\alpha_1 = \cos(V_a, \partial V_c) \quad 0.5 < \text{a} < 1.0 \tag{4}$$

where a is weight of source users. D is normal user, his weight is 1-a. So the function of calculating similarity between A and D as (4), but the weight factor a is replaced by 1-a.

But when calculating A and B's similarity of interests, we use cosine similarity without weight, because them are equal, and then we build tie-strength through their owns and common friends as (5)

Fig. 1 User's relationship model with common friends



$$Ti(A, B) = S + \sum_{i=1}^{n} \alpha_1 \alpha_2 \tag{5}$$

where α_1 is tie-strength between user i and A, $\alpha_2$is tie-strength between user i and B, n is total number of A and B's common friends.

3.3 Calculate Scores

At last we calculate the similarity between user and news as part of final score, using given user's interests vector and topic vector of recommending content, and tie-strength between users as another part of final score, we use weight α to adjust combination of two parts as (6). We recommend higher scores information to user.

$$Score(u, i) = \partial \cos(V_u, C_i) + (1 - \partial)Ti(u, B) \tag{6}$$

This calculates how user u is interested in new i. V_u is user u's interest vector, C_i is a set words of new i, Ti(u,B) is tie-strength between user u and B, which user B is the owner of new i.

4 Dataset and Experiment

In our research, to improve the precision of recommending, we use followees' interests to extend user's interests, we also use relationships between followees to measure tie-strength, so we compare the following five ranking algorithms for research:

- Default: When we use social network, we read from social streams, which contains information that posted by our followees. It's ranked by time that information is posted. We use this recommendation as baseline for other algorithms.
- Intelligent Ranking: Sina micro-blog has a recommendation system, which called Intelligent Ranking, it can rank information according users' interests.
- No followees' interests: When building user's interests, we don't use their followees' interests. But we use their common friends tie-strength to measure their tie-strength.
- No tie-strength: Building user's interest model, we use their followees' interests, but we don't use common friends' tie-strength to measure their tie-strength.
- With followees' interests and tie-strength: When building user's interests, we use their followees' interests. And we also consider tie-strength when recommending.

In our recommendation system, we should build users' interest model firstly. We use three factors: contents, tags and followees' interests. When users authorize to use our system, we get these information and construct their interests using above algorithms. Then we update their interests incrementally, we just need to analyze new parts, such as new followees and new statuses. We put this operation offline at midnight every day. When information comes, we have had users' interests already, we compute the similarity between user's interests and topic vector of information and tie-strength between user and owner of this information, and each information gets a score, we rank information according to these scores and recommend higher scores information.

When adding followees' interests to user's interest model, we need to distinguish followees. Different type of followees has different weight. We experiment Naive Bayes to make sure it can help us to divide them. We get 150 users' data from sina APIs, each user's data contains four elements: number of followees, number of fans, number of bi-followers, number of statuses, and choose 130 users randomly to compose train set, and rest 20 users compose test set. We use the train set to train the classifier, and get percentage of influence of each property, and then we use test set to test the accuracy of classifier. We repeated this operation ten times, and figure out average error rate, and we use the average percentage of each property in our recommendation system.

When constructing user's interests, we separate all contents into groups since users post first status, each group contains one month's contents. Short-term interests only use the last group, long-term interests use all contents exclude last month. So we can reuse short-term interests rather than analyzing all contents repeatedly. In particular, user's followees may be too many, which we are not able to process all of them. Sarwar et al. [16] have shown that by considering only a small neighborhood of people around the end user, we can reduce the set of items to consider, we'll remove some persons, who have statuses less than 20, this will reduce the amount of dataset effectively.

We deployed recommendations online for two weeks, and invited 52 active users to participate in our research. We divided these users into three groups, group one(G1) will use our system every day, group two(G2) use system every other day, group three(G3) use system every three day. Every time users came, we achieved first one hundred news in Sina Intelligent Ranking, and pick up fifty news from beginning, middle and end position, then we ask users to rate every new with five star according to their interests, which like Fig. 2. And we feed these fifty news to another four algorithms to rank them, and the five algorithms generate a total of 250 recommendations each time, every new has a score in each algorithm.

Then we use ASPT(average scores per time) to estimate the accuracy of each algorithm. Method of calculating ASPT as (7)

$$ASPT = \frac{1}{N} * \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} S_j P_j}{\sum_{i=1}^{N} S_{\max}} \tag{7}$$

where N is total numbers of groups, M is total numbers of each group, here we choose 50; S_j is score of new i that users rate. In each recommendation, new i is in P_i position. S_{max} is the max score that one micro-blog can get , we define it as five. If algorithm has higher ASPT, users are more satisfying with this algorithm.

Fig. 2 Online test website layout

5 Discuss

We ran our study online for two weeks and invited 52 real users to participate, each user produced one group score data each time of each algorithm, users in first group will totally produce 1,190 groups score data; users in second group will totally produce 595 groups score data; users in third group will totally produce 360 groups score data, and each group contains 50 scores, because each time we recommend 50 news.

Besides we have experiments for dividing followees types. We use 150 users' data for experiments for training and testing the classifier, and we repeat training for ten times, we get the influence of each property for different types of users.

In the following part, we will discuss our experiments and results from the followees division, user's interest model, tie-strength and stability of algorithms.

5.1 Followees Division

Figures 3 and 4 are the results of test of Naive Bayes. In each figure, every column represents one experiment, and x- axis is error rate of each test, y-axis is percentage of influence of each property.

Figure 3 represents information source users. From this figure, bi-followers and friends have lower influence, but fans and statuses hold higher percentage. Information source users always post news, many users follow them to get newer and

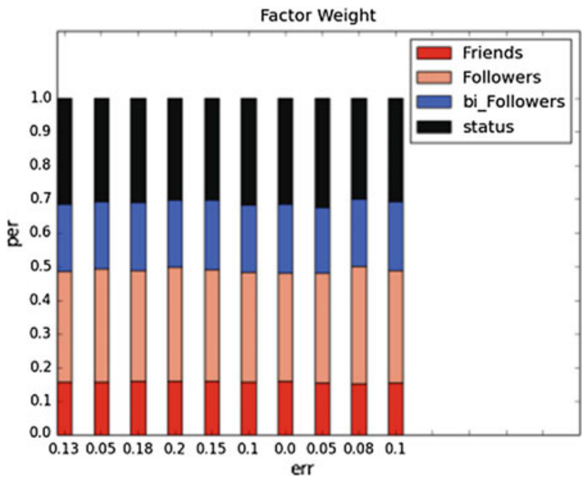


Fig. 3 Error rates of source users and influence percentage of each property

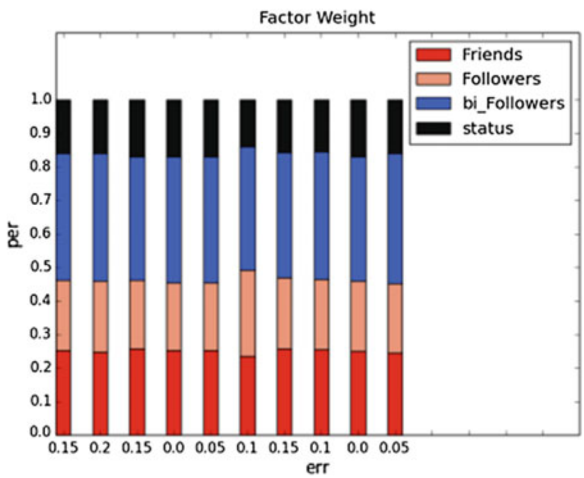


Fig. 4 Error rates of normal users and influence percentage of each property

valuable information. We found that bi-followers are mostly their followees, that's to say their followees mostly follow them back, because the information source users are more like to follow each other, so they can get more value information from each other.

Figure 4 represents normal users. From figure, fans and statuses don't get high percentage, especially statuses, because this type of users likes to read news, not to write, they want to get more information from others, so followees get a higher percentage. But bi-followers has highest percentage of influence, unlikely source users, normal users' bi-followers are mostly their fans.

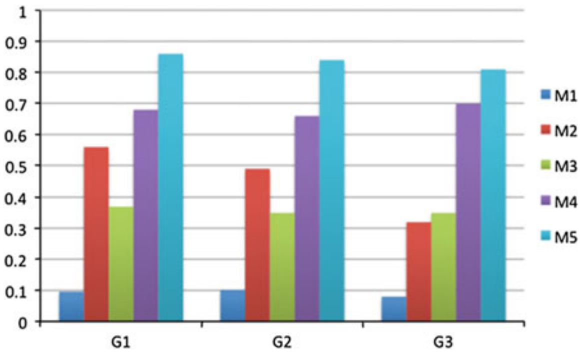


Fig. 5 Distribution of scores of each algorithm rated by each group

So we can use the four characters above to divide users' type, because information source users have more statuses and fans, and bi-followers are mostly their followees; normal users have little statuses and fans, and bi-followers are mostly their fans.

5.2 Interest Model and Tie-Strength

When volunteers use our system, we ask them to rate each news according their interests. At last we use these scores to compare each algorithm. The results as Fig. 5.

To reduce the complexness of figures, we give every algorithm an abbreviation. We call the default algorithm as M1; Sina Intelligent Ranking as M2; No followees' interests as M3; No tie-strength as M4; and with followees' interests and tie-strength as M5.

From Fig. 5, M1 has lowest degree of satisfaction, because M1 ranks by time, so user should make much time to filter information; the others consider user's interests and contents, they get higher scores, and our recommendation is better. As we mentioned earlier, tags and contents can't always build our interest model precisely due to low activity, so M3's performance is lower than Intelligent Ranking. When considering followees' interests, M4 and M5 have higher precision, because through their followees' interests, we extend users' interests. If we consider tie-strength between users, the satisfaction of recommending improved about 12 %. So tie-strength is an important factor for recommending.

5.3 Stability of Algorithms

From Fig. 6, When user don't get in website for a long time, Intelligent Ranking performance begins to decline, We find that Sina Intelligent Ranking recommends information that is mostly posted by who we always prefer to read from, it mays

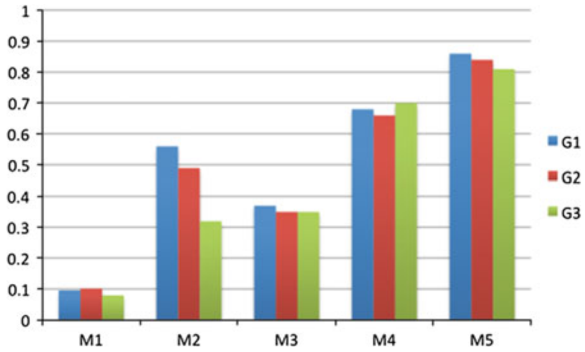


Fig. 6 Distribution of scores from another view

record these persons, who we always read news from, not our interests, so it's not precise when we rarely use it.

However, we construct users' interests from long-term and short-term interests, through this way, users' interest model won't change rapidly, so our algorithms are more stable.

So When recommending news, we consider users' long-term and short-term interests to improve the stability of algorithm, and construct users' interests with their followees' interests to improve the precision of recommending.

6 Conclusion and Future Work

In this paper, we have studied content recommendation on sina micro-blog. We implemented three algorithms. We compared each algorithm's performance with Sina' Intelligent Ranking algorithm through the usage results of 52 sina users, we found that our algorithm, which consider user's long-term and short-term interests and tie-strength between users, are more precise and stable than Intelligent Ranking algorithm.

As mentioned above, there is more implicit information than explicit. In our research, we have used users' behavior of following, content and tags. Future research may add more dimensions to construct user's interests and tie-strength. People use micro-blog for different purpose, such as information purpose and social purpose [4], but in our research we don't distinguish these too much. We'll pay more attention to recommendation for different usage purposes in our future work.

In our ranking algorithm, although we consider the importance of individuals is different, in social network, there are many areas of interest, but we regard every user from different areas of interest as same. In future research, we will refine grain size, and divide people into more different groups according their interests, and choose the highest words from each group.

References

1. Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
2. Bernstein, M., Suh, B., Hong, L.: Eddi: Interactive topic-based browsing of social status streams. In: Proceedings UIST'10. 2010
3. Boyd, D., Golder, S., Lotan, G.: Tweet tweet retweet: Conversational aspects of retweeting on Twitter. In: Proceedings HICSSS'10, 2010
4. Chen, J., Nairn, R., Chi, E.: Speak little and well: recommending conversations in online Social streams. In: Proceedings CHI'11. pp. 217–226. ACM, New York, USA (2011)
5. Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E.H.: Short and tweet: experiments on recommending content from information streams. In: Proceedings CHI'10, ACM Press (2010)
6. Gilbert, E., Karrahalios, K.: Predicting tie strength with social media. In: Proceedings CHI'09. 2009
7. Granovetter, M.S.: The strength of weak ties. Am. J. Sociol. **78**(6), 1360–1380 (1973)
8. Kincaid, J.: EdgeRank: The secret sauce that makes Facebook's news feed tick. Retrieved from http://techcrunch.com/2010/04/22/facebook-edgerank 2010
9. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Comput. Soc. **7**(1), 76–80 (2003)
10. Mehrotra, R., Sanner, S., Buntine, W., Xie, L.: Improving LDA topic models for Microblogs via tweet pooling and automatic labeling
11. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of ACM DL'00. pp. 195–204 (2000)
12. Paek, T., Gamon, M., Counts, S., Chickering, D.M., Dhesi, A.: Predicting the importance of newsfeed posts and social network friends. In: Proceedings AAAI'10. 2010
13. Pazzani, M.J., Muramatsu, J., Billsus, D.: Syskill & webert: Identifying interesting web sites. pp. 54–61, AAAI/IAAI (1996)
14. Sahami, M., Heilman, T.D.: A Web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of WWW'06. 2006
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process Manage. **24**(5), 513–523 (1988)
16. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Recommender systems for large-scale E-Commerce: Scalable neighborhood formation using clustering. In: Proceedings of ICCIT'02. 2002
17. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: proceedings of SIGIR'02. 2002
18. Weng, J., Lim, E., Jiang, J., He, Q.: TwitterRank: finding topic- sensitive influential twitterers. Proceedings of the 3rd ACM international conference on web search and data mining (WSDM □ 10), pp. 261–270. ACM, New York, USA (2010)
19. Yu, Y., Qiu, G.H., Chen, A.P.: Friend recommendation algorithm based on mixed graph in online social networks. New technology of library and information service. 2011