

# Friend Recommendation System Based on Geolocation and Contents of Users

Mingyue Duan, Jing Huang\*, Jiaxuan Zhang, Yunjie Tian, Dongwei Li and Bin Wang

College of Computer Science and Technology, Jilin university, Changchun 130012

Key Laboratory of Symbol Computation and Knowledge Engineering (Jilin university), Ministry of Education, Changchun 130012

\*Corresponding author

**Abstract**—In recent years, with the rapid development of wireless mobile network and smart phone operating systems, various social software based on wireless Internet has emerged one after another. Current popular social software such as QQ and WeChat have become important tools for people to meet new friends. At present, existing social software can recommend other users in the vicinity according to the geographical location of the user. However, this method does not consider the user's interests, hobbies, etc. So that the effectiveness of such a friend recommendation system is often unsatisfactory. In order to solve the above problems, a personalized friend recommendation system based on geolocation information and user content is designed and developed. In this system, not only the geolocation information of the user is considered, but also the features of the user's published statuses are extracted, aiming to recommend more similar other users to the user. After testing, the effectiveness of the proposed method is verified.

**Keywords**—instant message; friend recommendation; geolocation information; context information

## I. INTRODUCTION

### A. Related Work

With the development of modern information technology, mobile communication has entered the 4G era. The development of wireless mobile communication has continuously increased the transmission rate and quality of the network, which greatly improves people's daily life experience. Mobile Internet terminals represented by smartphones and tablets have continuously improved their performance. They have become an important means for people to access the Internet and experience mobile services in everyday life. Instant messaging software has become one of the essential communication tools in life because of its immediacy and convenience<sup>[1-8]</sup>.

Within the scope of the mobile Internet, Tencent QQ and WeChat are very successful instant messaging software and have shown great dynamism on the mobile phone, which has been highly praised by users. Among them, QQ, as a traditional chat tool, has realized a wealth of instant messaging functions, such as improving pictures, voice intercom, and video calls, and integrating various social functions such as QQ mailbox, friends' dynamics, and introducing people they may know. WeChat as a rising star of instant messaging software, also integrates some relatively new social functions such as Wechat Moments and people nearby.

Although the social software mentioned above is very powerful in function, the homogeneity of its products is pretty serious, and there are also disadvantages in that the user's positioning is not clear enough. Current domestic and international social software and strangers make friends based on the same place at the same time. For example, by "people nearby" on WeChat, users can make some friends outside of their own real life in this way. For example, "people nearby" on WeChat, users can make some of their own real life in this way friend. In order to improve the online dating rate of online users, some online users who are geographically arranged from near to far are displayed in WeChat "people nearby". However, in real life, we will consider a lot of information in the process of finding a friend, such as the other's age, occupation, interests, hobbies and so on. After investigation, we found that people with more similar interests, hobbies and similar experiences are more likely to become friends. The current existing friend recommendation algorithm only recommends based on the user's geolocation information without considering the user's mood, state, interest at that time and hobby, etc. For example, in a university, a student could find people who are engaged in a series of work such as students, teachers, cleaners, cooks, etc on his WeChat "people nearby". Obviously, if the student has added cleaners, chefs or others who have any other job as his friends, there will be no common topic between them. Therefore, such a recommendation is undoubtedly unreasonable. Based on the above situation, we have designed and developed a new type of social software. In this software, we can not only show the current user who people nearby are, but also perform historical data-published status to the users and do the feature extraction. Then, the similarity calculation is performed using the extracted features and other users on the nearby network. Finally, other users are displayed to the user according to the degree of similarity.

### B. Main Contributions

The main contributions of this paper are as follows:

We have designed and implemented of a personalized friend recommendation system based on geolocation information and user content information, which is different from the existing social software that only uses time and geolocation information to recommend friends. Our system is based on the current time and geolocation information to find other nearby users, then use *TF/IDF* algorithm to carry on the keywords of the statuses that these users published before, and

finally recommend them to the user according to the user's own characteristics. The effect is good through the use of 4 classes.

### C. Organization Structure

The remainder of this paper is organized as follows. Section 2 briefly the design of friend recommendation system. Section 3 presents the recommendation algorithm model in detail. Experimental evaluation is shown in Section 4. Finally, Section 5 concludes the paper with a brief discussion.

## II. THE DESIGN OF FRIEND RECOMMENDATION SYSTEM

### A. System Requirements Analysis

According to the needs of mobile phone users for instant communication through wireless terminals such as mobile phones, we analyze the principle of the XMPP protocol, the principle of the RTP/RTCP protocol, and use the C/S software architecture. The specific functions of the software are as follows:

1) *Registration*: The user enters the mobile phone number to register. If the number has not been registered yet, the user can complete the registration by filling in the password; otherwise, it indicates that there is currently a user.

2) *Login*: The user enters the user name and password. If the cloud server authentication succeeds, the login succeeds and the system jumps to the main interface. Otherwise, the login fails.

3) *Adding friends*: The user can find other users according to the user name, and send a request for adding a friend. If the other user agrees, the cloud server makes a corresponding request processing, and shows that the adding is successful.

4) *Deleting friends*: Users can delete friends and make corresponding processing on the cloud server.

5) *Chatting*: Users can initiate chats and the chatting records are saved to the local and cloud at the same time.

6) *Posting statuses*: Users can publish statuses, and the system will automatically save statuses, time information, and geolocation information to the cloud server.

7) *Deleting statuses*: the user can delete the statuses and perform corresponding processing on the cloud server at the same time.

8) *Viewing "people nearby"*: When the user clicks on a button named nearby person, the system will automatically search based on time and geolocation information to find other users who are online at the same time. Then calculate the similarity of the user's feature vectors which are calculated by the natural language processing algorithm. The calculation is based on the cosine similarity<sup>[9]</sup>. Then the result is displayed to the current user in terms of the size of the similarity.

9) *Positioning*: the system needs to monitor the user's position in real time

### B. Overall Design of the Friend Recommendation System

The software is suitable for all kinds of users. The client and server use TCP connection to realize instant message among users. Therefore, we have realized the software design of the server and the client respectively. Among them, the

cloud server is responsible for listening to the user's connection request, storing the user information in the linked list, sending the friend information to the user, recording the statuses sent by the user, and screening and pushing the user's statuses. The client initiates an active link and requests or registers with the server. In the client, the user can input his own user name and avatar. The client can initiate P2P and multiplayer chat and apply to the client to add and delete friends. The user can also send statuses on the client and accept the processed push statuses. The server and client use cases are shown in Figure 1.

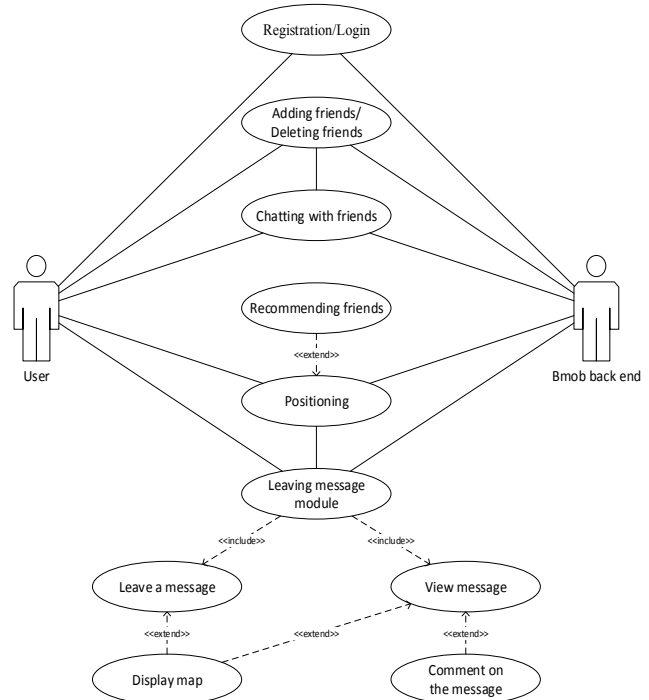


FIGURE 1. THE SERVER AND CLIENT USE CASE DIAGRAM

## III. RECOMMENDATION ALGORITHM MODEL

### A. Basic Concepts

The content-based recommendation algorithm mainly constructs the user's preference document based on historical information (eg. evaluation, sharing, and favorite texts), and calculates the similarity between the projects and the user status documents, and then recommends the most similar item to the user. In this article, historical information refer to the statuses published by users. According to the user's published statuses, the user's status document is constructed. Then the keywords are used to represent their features, then the *TF/IDF* (term frequency-inverse document frequency)<sup>[10]</sup> is used. The algorithm determines the weight for each feature. After argument, it is also a very effective algorithm. Finally, calculate the similarity between users and recommend them to users in descending order of similarity.

### B. TF/IDF Algorithm

*TF/IDF* is an algorithm for calculating text features proposed by Salton and others in 1973. This algorithm is often used to calculate information retrieval and data mining weights. The basic idea of this method is: On the one hand, the more

times the keyword  $k$  appears in the user's status document  $D$ , and the fewer times of occurrences of the keyword in other user status documents, the more the keywords can be used to represent the feature of the user; on the other hand, the more times a keyword appears in different documents, the less contribution it make to the difference document.

Before the *TF/IDF* algorithm, we need to do two things: first, perform text segmentation on the user status document, that is, split the user's statuses into a combination of individual words; second, delete the stopwords in the user's documents. The so-called stopwords are those words that do not contain actual meaning, such as "I", "is", "this", etc. It can be considered that these words cannot represent the content of the texts. After obtaining the keywords for each user's status document, we can calculate the feature weight vector for each user's status document.

In the *TF/IDF* algorithm, term frequency (TF) refers to the number of times a certain keyword appears in the user's status document  $d_j$ . If there is a user's status document  $d_j$  with a keyword  $k_i$  in the document, the term frequency of  $k_i$  in the document is defined as:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

Among them,  $n_{i,j}$  represents the number of times the keyword  $k_i$  appears in the document  $d_j$ , and  $k$  represents the keyword appearing in the document  $d_j$ . Inverse document frequency (IDF) indicates the importance of the keyword  $k$  in the document set. The basic idea is that the fewer documents containing the keyword  $k$ , the stronger discrimination ability of  $k$  is. The inverse document frequency of the keyword  $k_i$  appearing in the user document is defined as follows:

$$IDF_i = \log \frac{N}{n_i} \quad (2)$$

Where  $N$  is the number of user status documents. The  $m$ -dimensional vector  $\mathbf{d}_j = (w_{1j}, w_{2j}, \dots, w_{kj})$  is used to represent a user's status document and  $k$  is the number of keywords. Each component in the vector can be calculated as follows:

$$w_{ij} = TF_{i,j} \times IDF_i = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \log \frac{N}{n_i} \quad (3)$$

At this point, we can obtain the text features of each user. Then calculate the similarity between users based on cosine similarity. Assuming that the text feature vectors of user A and user B are  $\mathbf{d}_A = (w_{1A}, w_{2A}, \dots, w_{kA})$  and

$\mathbf{d}_B = (w_{1B}, w_{2B}, \dots, w_{kB})$  respectively, the similarity of these two users can be calculated as follows:

$$\text{similarity}(A, B) = \frac{\sum_{i=1}^k w_{iA} w_{iB}}{\sqrt{\sum_{i=1}^k w_{iA}^2} \cdot \sqrt{\sum_{i=1}^k w_{iB}^2}} \quad (4)$$

#### IV. SPECIFIC CALCULATION PROCESS AND RESULT ANALYSIS

In our friend recommendation system, in addition to screening the user within a certain range of the current user's current geolocation information, we also use the user's published statuses as the user's main features. Context-based recommendation takes into account the nature of the object itself. In general, the statuses published by the user contains a lot of information of the user. One way to compare the texts using *TF/IDF* is to compare the keywords, that is, to extract several keywords in each statuses, that is, in the text, and then use these keywords as characteristics of this paragraph of text. Assume that the users U2 and U3 are near the user U1 and the geographical distances are 200m and 500m respectively. If only the friend is recommended based on the geolocation information, the recommendation sequence should be U2→U3. Using our method to recommend friends is calculated as follows: The three users have published many statuses in the past. We use the two most recent statuses as the content of the user's status document. The contents are as follows:

U1:

Today, I was very happy to have dinner at Rixin Building!

Haha, I ate in the cafeteria at noon and I was met with a confession!

U2:

Today, knowledge the teacher taught was very difficult! Depressed,

The homework is so perverted that I hate this teacher.

U3:

Tomorrow I will date my girl. Everyone will cheer me on. Please eat!

Confess success! I'm going to eat delicious food.

Based on the published statuses of these three users, we recommend friends to U1 in descending order of similarity. According to the *TF/IDF* algorithm, first, the statuses of the three users is decomposed and the stopwords are deleted, then the similarity between these keywords is calculated. The statuses decomposition results of the users U1, U2, and U3 are shown in Table 1, Table 2 and Table 3.

TABLE I. STATUSES DECOMPOSITION RESULT OF U1

keywords	$TF_i$	$IDF_i$	$w_{i1}$
Rixin Building	1/8	1.0986	0.1373
eat	1/4	0.4055	0.1014
happy	1/8	1.0986	0.1373
noon	1/8	1.0986	0.1373
cafeteria	1/8	1.0986	0.1373
met	1/8	1.0986	0.1373
confess	1/8	1.0986	0.1373

TABLE II. STATUSES DECOMPOSITION RESULT OF U2

keywords	$TF_i$	$IDF_i$	$w_{i2}$
Teacher	2/9	1.0986	0.2441
taught	1/9	1.0986	0.1221
knowledge	1/9	1.0986	0.1221
difficult	2/9	1.0986	0.2441
depressed	1/9	1.0986	0.1221
homework	1/9	1.0986	0.1221
hate	1/9	1.0986	0.1221

TABLE III. STATUSES DECOMPOSITION RESULT OF U3

keywords	$TF_i$	$IDF_i$	$w_{i3}$
confess	1/10	0.4055	0.1014
success	1/10	1.0986	0.1099
eat	3/10	0.4055	0.1216
girl	1/10	1.0986	0.1099
date	1/10	1.0986	0.1099
everyone	1/10	1.0986	0.1099
cheer	1/10	1.0986	0.1099
please	1/10	1.0986	0.1099

Substituting  $k = 5$  into formula (4), the result obtained after calculation is:

$$\text{similarity}(U1, U2) = 0.9439$$

$$\text{similarity}(U1, U3) = 0.9991$$

$$\text{similarity}(U2, U3) = 0.9515$$

According to the calculation, the similarity between the user U1 and the user U3 is relatively large, so the recommendation sequence should be  $U3 \rightarrow U2$ . In normal logical thinking, we can find that users U1 and U3 have published terms such as "confess" and "eat", and the statuses published by user U2 is obviously something related to "learning". This is consistent with our calculated results.

## V. CONCLUSION

We have designed and developed a personalized friend recommendation system based on geolocation information and statuses users published. Unlike the current method of recommending friends based only on geolocation information, we have added information on the statuses of users based on geolocation information. To a certain extent, it makes up for the lack of methods of friend recommendation in social software, and has very good practical value and a certain degree of innovative significance. Through the use of 4 classes, the effect is good.

## ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China under grants 61373053 and 61572226, and Jilin Province Key Scientific and Technological Research and Development project under grants 20180201044GX and 20180201067GX.

## REFERENCES

- [1] Friendship and mobility:user movement in location-based social networks. Cho E,Myers S A,Leskovec J. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining . 2011
- [2] From instant messaging to cloud computing, an XMPPreview. Hornsby A,Walsh R. 2010IEEE14th International Symposium on Consumer Electronics . 2010
- [3] Secured video streaming development on smartphones with Android platform. MASSANDY D T,MUNI I R. Proceedings of 2012 7th International Conference on Telecommunication Systems,Services,and Applications . 2012
- [4] Application of XMPP Protocol in Android Instant Messaging System [J]. Li Xinlu. Computer Knowledge and Technology. 2013(28)
- [5] TCP connections for P2P apps:A software approach to solving the NAT problem. Jeffrey L.Eppinger. Technical Report CMUISRI-05-104,Carnegie Mellon University . 2005
- [6] Design of instant messaging system based on XMPP protocol for Android [J]. Huang Weimin. Electronic Design Engineering. 2011(08)
- [7] Exploring Chinese users' acceptance of instant messaging using the theory of planned behavior, the technology acceptance model, and the flow theory[J] . Yaobin Lu,Tao Zhou,Bin Wang. Computers in Human Behavior . 2008 (1)
- [8] Perceived academic effects of instant messaging use[J] . Reynol Junco,Shelia R. Cotten. Computers & Education . 2010 (2)
- [9] Cosine similarity measures for dual hesitant fuzzy sets.Wei Qiu,Guangchun Zhang,Lei Zhu.2016 4th International Conference on Machinery,Materials and Computing Technology(ICMMCT 2016)
- [10] TF-SIDF:Term frequency,sketched inverse document frequency. Baena-García M,Carmona-Cejudo JM,Castillo G,et al. 2011 11th International Conference on Intelligent Systems Design and Applications (ISDA) . 2011