A Generic API for Retrieving Human-oriented Information from Social Network Services

Teruaki Yokoyama, Shigeru Kashihara, Takeshi Okuda, Youki Kadobayashi, and Suguru Yamaguchi
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
{terua-yo, shigeru, okuda, youki-k, suguru}@is.naist.jp

Abstract

A unique type of Web service, called a Social Network Service (SNS), first appeared in 2003. Some researches suggested a method to extract meaningful information from SNSs. Such meaningful information helps computers to understand human interests, concerns, and perceptions. We consider that socially aware computing has the potential to lead a new type of metric such human oriented information to access control, searches, and suggestions for contents. Accordingly, in this paper we propose the access Application Programmable Interface (API) to find human-oriented information from outside a given SNS. We use the result of experiments on the API to demonstrate a network comprising relationships among communities. The results reveal that our proposed method can support developers in creating socially aware applications with SNSs.

1 Introduction

A unique type of Web service, called a Social Network Service (SNS), first appeared in 2003, and SNS sites are now widely used by Internet users all over the world. *Friendster* [7] and *Orkut* [10] are regarded as the pioneers of SNS, and they have become representative SNS sites. Meanwhile, *Mixi* [9], a domestic SNS site in Japan, has succeeded in obtaining more than five million users. SNSs are special in that only people who are invited by current members can join a given service. SNS users build relationships among themselves and become members of circles, which are called $communities$, and SNSs record much information from such users' activities. While one related work [13] remarks that we can extract human-oriented information from SNSs, at present current SNSs only act as communication platforms for recreation.

The aim of socially aware computing [11] is to create applications that understand human circumstances such as favorites, intersts, concerns and perceptions. Developers of socially-aware computing need large amount of information about human behaviors to ensure that computers have sufficient data on human circumstances with which to operate. Note that we call the information about human circumstances "human-oriented information," and socially aware computing is defined as a computing style based on human-oriented information. A lack of such human-oriented information can cause undesired communication on the Internet, e.g., SPAM, SPIM and SPIT [3]. Moreover, insufficient human-oriented information makes it difficult for people to find desired information from the glut of it on the Internet. We consider socially aware computing to be a potential solution to such problems on the Internet.

Some studies have concluded that it is possible to find human relationships based on the human-oriented information from people's behavior extracted from the enormous archive of digital data such as exchanged e-mail, mailing list archives, and Web pages [1][2][5]. We consider that SNSs can also be a good repositories of human-oriented information because it includes explicitly constructed human relationships, recorded human behaviors through their communication. Obtaining both a large quantity and variety of information is important for socially aware computing since it results in more accurate computing. However, current SNS does not provide any method to show the contained information. Based on this concept, we consider a method to allow people to use human-oriented information in SNS.

In this paper, we propose an SNS API (Application Programmable Interface) to obtain human-oriented information from outside an SNS system. We implement the SNS API with the aim of providing access functions in *Mixi*. We have selected *Mixi* as the example case, and demonstrate using our API. The result is finding relationships among communities from the user's point of view. The advantage of our API is that it enables all users to extract such results, since theoretically our API is applicable to common SNSs as well as *Mixi*. Furthermore, our proposed API contributes to helping developers to create socially aware applications with SNS. We believe that SNS would be human relationship repository for socially aware computing.

2 A Proposal for Access API to SNS

As stated in Section 1, we consider at SNS to be an important communication platform, i.e., an SNS is a database system for human-oriented information. In this section, we first describe an outline of SNS, then propose an access API to SNS for SNSs to extract human-oriented information from them. Our API enables every user to extract human-oriented information including community relationships in SNSs.

Proceedings of the 2007 International Symposium
on Applications and the Internet Workshops (SAINTW'07)
0-7695-2757-4/07 $20.00 © 2007 IEEE

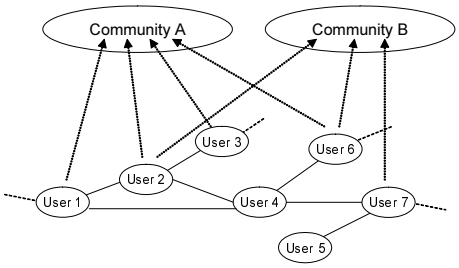

Figure 1. Graph structure inside SNS

2.1 Social Network Service

To join in an SNS, non-members needs to receive an invitation message from a current member. Accordingly, when non-members join an SNS, they establish relationships with current members. Moreover, they can freely take part in various communities, and can participate in discussions and exchange information in the community. The relationships among members and/or communities in an SNS are explicitly expressed as digital data; that is, we can say that an SNS is considered as an important repository for storing human activities.

Figure 1 depicts an example of a network structure among members and community groups. All users on the SNS can freely create favorite communities as their communication space. A user who is interested in the community can take part in it, and most users tend to join in multiple communities. When it comes to the communities, they are considered as groups of people holding similar interests; therefore, the SNS can display the structure between communities and users as well as show the user's relationships.

2.2 L2-Norm for Extracting Community Relationship

Here we define measures to calculate relationships between communities. E. Spertus, et al., showed that L2-Norm is the best measure of similarity between communities in an SNS from empirical results [13]. We have selected the L2-Norm for extracting community relationships in our API.

The following is an explaination of the method for extracting community relationships with the L2-Norm. As Fig. 1 shows, the relationships between users and communities are displayed as a bipartite graph. Users joining multiple communities indicates that the communities may have similarity because the propensity for each user is converged. Viewed in this light, we define the popularity between two communities $P_{i\to j}$ as Formula 4, which means the ratio of the relationship from community i to community j.

Figure 2 illustrates the model of community relationships in an SNS, where C_i denotes a set of participants in community i, $n(X)$ is a function to count the number of elements in target set X, M_i represents the number of participants in community i, M_{ij} indicates the number of participants joining in both of communities i and j, and C_{ij} is a set of users joining in both of community i and j. The ratio $P_{i\to j}$ approaches 1 when a large number of users in community i are interested in community j. Namely, in this case,

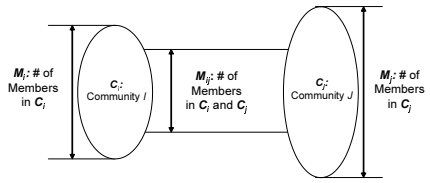

Figure 2. Relationship model in SNS

almost all the users in community i are completely included in community j. On the other hand, the ratio approaches 0 if there is no overlap of users in both communities. If that happens, we employ the unidirectional correlation (Formula 4) as the measure for extracting community relationships.

$$C_i \in \{members\ of\ community\ i\} \tag{1}$$

$$M_i = n(C_i) \tag{2}$$

$$C_{ij} = C_i \cap C_j \tag{3}$$

$$P_{i\to j} = \frac{M_{ij}}{M_i} \tag{4}$$

$$R_{ij} = \sqrt{P_{i\to j} P_{j\to i}} = \frac{M_{ij}}{\sqrt{M_i M_j}} \tag{5}$$

We employ the Formula 5 for calculating the bidirectional correlation between communities. This is known as the L2-Norm measures. From Formulas 1 and 3, the measures only needs participation information of two targeted communities, C_i and C_j. Then, we can calculate R_{ij} from such local information around target communities.

2.3 API Functions to extract human-oriented information

We propose a generic API for retrieving human-oriented information from an SNS, and describe the representative functions here. These semantics are available for every SNS.

1. **Login(**ID**,** $Password$**)**:
   This function is used to gain access permission to the target SNS before using our functions.
2. **Communities(**UID**)**:
   This function returns the participation information of the target user specified from the User ID (UID).
3. **Members(**CID**)**:
   This function returns the participant's information about the target community from the Community ID (CID)
4. **CalcSimilarity(**CID_A**,** CID_B**)**:
   This function extracts the similarity between communities A and B. The calculation is based on L2-Norm measures described in the previous section.
5. **RCommunities(**CID**,** n**)**:
   This function returns the n communities that are closely related to target community CID. (See algorithm 2.1)

Proceedings of the 2007 International Symposium
on Applications and the Internet Workshops (SAINTW'07)
0-7695-2757-4/07 $20.00 © 2007 IEEE

6. **CommunityNetwork(**CID**,** n**,** $dist$**)**:
   This function extracts the relationship network from target community CID. The function begins to search for a relationship from community CID and then it finds closely related n communities iteratively. The process is repeated $dist$ times. Finally, the function returns extracted community relationships as graph structure G. Algorithm 2.2 describes the iteration process.

Functions 1 to 3 provide fundamental services in an SNS, e.g., finding type of information about other users and communities. Functions 4 to 6 provide the method of extracting the relationship between the communities. These relationships are calculated in local computer based on the information from three fundamental functions. Furthermore, functions to extract another information from an SNS can be easily added to our API.

Algorithm 2.1: RCOMMUNITIES(CID, n)

comment: Extract n related communities with CID

$UIDs \leftarrow$ MEMBERS(CID)
$CIDs \leftarrow$ COMMUNITIES($\{u | u \in UIDs\}$)
for each $c \in CIDs$
  do $Rel[c] \leftarrow$ CALCSIMILARITY(CID, c)
sort Rel in descending order
return (first n of Rel)

Algorithm 2.2: COMMUNITYNETWORK($CID, n, dist$)

comment: Extract community network structure

$CIDs \leftarrow CID$
for $i \leftarrow 1$ **to** $distance$
do {
  for each $c_1 \in CIDs$
  do {
    $Rel \leftarrow$ RCOMMUNITIES(c_1, n)
    for each $c_2 \in \{x | Rel[x] > 0\}$
    do {
      $V \leftarrow V \cup c_2$
      $E \leftarrow E \cup (c_1, c_2)$
      $CID_r \leftarrow CID_r \cup c_2$
  $CIDs \leftarrow CID_r$
$G = (V, E)$
return (G)

3 Demonstration and Discussion

We have implemented our proposed API to obtain meaningful data from *Mixi* based on the concept in described in Sec.2, employing the program language *ruby* [12] to implement the API and the visualization program *graphviz* to illustrate relationships. In this section, we provide a demonstration in which we use our API to visualize relationships among communities. A figure is used to clearly reveal the semantic structure of community relationships.

Table 1. Five closely related communities to the "Jazz" community ($CID = i = 109, M_{109} = 3539$)

	j	Community name	M_j	$M_{109\,j}$	$R_{109\,j}$
1	10251	Miles Davis	1268	412	0.1944
2	9449	Herbie Hancock	821	299	0.1754
3	10216	John Coltrane	824	297	0.1739
4	1366	Bossa Nova	1897	429	0.1655
5	6922	Bill Evans	610	243	0.1653

3.1 Extracting communities relationship with "Jazz"

We select the "Jazz" community, which was created for talking about jazz music, as an example of our demonstration. The Jazz community has 3539 participants. Table 1 shows the five most closely related communities within the community "Jazz," with tha table sorted by correlation rate R_{109j}. The first column shows the ranking of the overlap rate. Consecutive columns are the community ID, community name, the number of participants, the number of duplicated members between each community and the community "Jazz," and the correlation rate.

We can automatically detect the communities that are closely related to the "Jazz" without understanding jazz music. Four of them are communities about jazz artist. The last one "Bossa Nova" is a style of Brazilian music. This result reveals that jazz music has a relationship with another genre of music – bossa nova – suggesting that people who love jazz music also like bossa nova.

Then, based on this result, we draw the network of community relationships as shown in Fig. 3. Starting from the community "Jazz," we apply our function as **COMMUNITYNETWORK("Jazz", 5, 3)**, consequently constructing community relationships with jazz music. In the figure, we can find three branches extending from the "Jazz" community. They are separated into clusters above and below the circle for the "Jazz" community, as well as a small one between them. The upper cluster includes bossa nova-related and similar communities to "Jazz". while the lower one includes jazz-related communities for discussing about jazz artists and playing styles of jazz. The small one between the two main clusters shows a remarkable result because the first community branching from the jazz and bossa nova clusters is a cluster of communities for people who loves both of jazz and bossa nova music. The name of this community is, interestingly, "quiet lovers with jazz and bossa nova".

3.2 Discussion

Our API was able to draw the community relationship with "Jazz," indicating that our API user can utilize an SNS as human-oriented information repository. However, some problems still remain. One of them is the issue of user's privacy: some people do not want to disclose their hobbies and

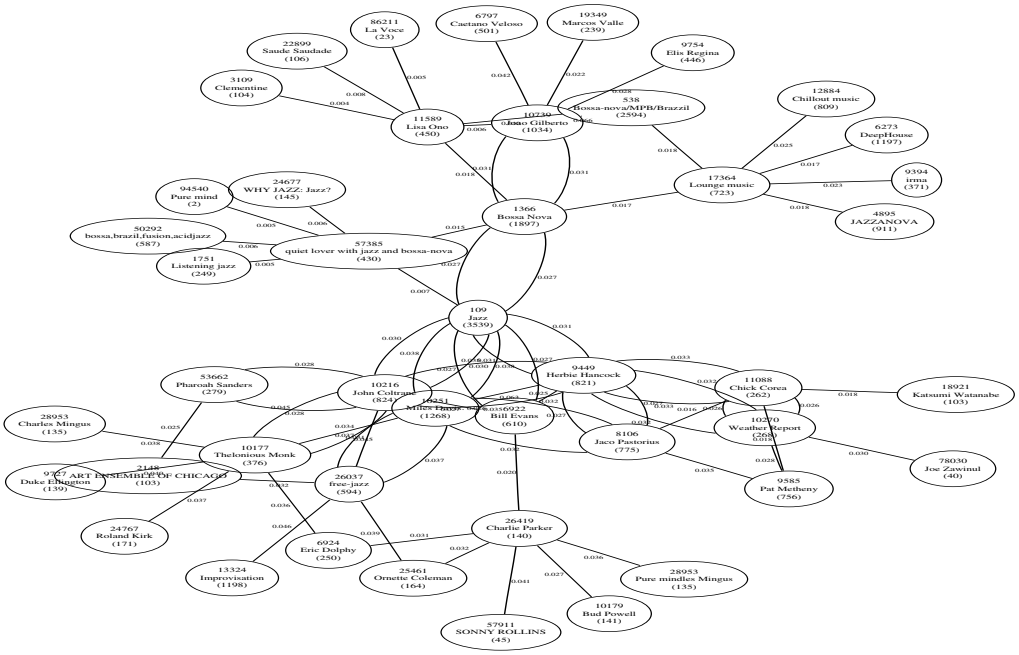

Figure 3. A Graph of Communities Relationship with Community "Jazz" $(CID : 109, n(C_{109}) = 3539)$

favorites to the public even if such information is opened on a conventional SNS.

Recent works [4][6] described how to obtain Social Network metrics for detailed behavior in daily life. such as talking and physical contact. They proposed methods for constructing SNSs dynamically. Furthermore, a new type of SNS has been proposed that aims to find users engaged in some special field like music, photography, and bookmarks[8]. SNSs are likely to diversify to detect a broad variety of many metrics in various propensities of users. We expect that such SNSs will be better repositories for human-oriented information if they adopt our proposed access API.

4 Conclusion and Future Work

In this paper, we proposed an access API to retrieve human-oriented information from Social Network Services. As a simple demonstration, we extracted "Jazz" related communities from SNS with our API. For example, people can receive e-mail only from persons who have similar interestings using the result. Moreover, people can find information from the persons.Like the example, SNS API allows computers to know social awareness. Using our proposed API, developpers will be able to implement social awareness onto their products.

In future work, we will enhance our API to be able to access not only to *Mixi* but to other SNSs. Moreover, we will construct some applications based on SNS information as examples of socially aware computing. As a result of concluding this work, we believe that the next generation of SNSs will serve as repositories of human-oriented information.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, July 2003.

[2] L. A. Adamic, O. Buyukkokten, and E. Adar. A social network caught in the Web. *First Monday*, 8(6), June 2003.

[3] V. G. Cerf. Spam, spim, and spit. *Commun. ACM*, 48(4):39–43, 2005.

[4] T. Choudhury and A. Pentland. Sensing and Modeling Human Networks using the Sociometer. In *Proc. of the International Conference on Wearable Computing*, October 2003.

[5] A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the Web. In *Proc. of First CEAS*, July 2004.

[6] N. Eagle and A. Pentland. Social Network Computing. In *Proc. of The Fifth International Conference o Ubiquitous Computing*, pages 289–296, October 2003.

[7] friendster. `http://www.friendster.com/`.

[8] M. Hamasaki, Y. Matsuo, K. Ishida, T. Hope, T. Nishimura, and H. Takeda. An integrated method for social network extraction. In *Proc. of the 15th WWW conference*, May 2006.

[9] mixi. `http://www.mixi.jp/`.

[10] orkut. `http://www.orkut.com/`.

[11] A. Pentland. Socially aware, computation and communication. *IEEE Computer*, 38(3):33–40, March 2005.

[12] Program language Ruby. `http://www.ruby-lang.org/`.

[13] E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating Similarity Measures: A Large-Scale Study in the Orkut Social Network. In *Proc. of the eleventh ACM SIGKDD*, pages 678–684, August 2005.