An Object-Oriented Social Networking to Link People with Similar Interests and Activities

Ching-Cheng Lee, Prachi Garg
Department of Computer Science, California State University at East Bay, Hayward, California, USA

Abstract - Social Networking sites have become increasingly prevalent in the society as a medium to socialize and connect with friends. These websites have changed the way we communicate and share our thoughts and experiences with other people. However, most social networking websites lack in connecting people over common activities or interests. Users often befriend other users with the objective to swap profiles and increase their friend list. Overtime, these connections become meaningless since there is no common interest or activity to bind them. This research proposes an effective approach to find users’ interests and suggest friends by implementing object-based features in a social networking application. Object centered sociality is suited for most social networking sites since it provides users a means to connect with other users who have similar level of interest.

Keywords: MyBook, MVC, JSP, Apache Tomcat, J2EE

1 Introduction

Most social networking sites focus primarily on building friends list and are more useful for contacting and locating old friends or make new ones. They allow users to create profiles, swap messages, and share photos – all with the goal of expanding their circle of online friends. These websites are good at representing links between people, but it does not explain what connects those particular people and not others [1]. They provide little attraction for repeated visits and therefore after some point become boring. Connecting content and people would provide a meaning to social networking [2].

1.1 Object-Oriented Sociality

Social networking sites provide features to find people with similar interests. However, they only use interests that are already specified by users in their profile and therefore are very static in nature. Often times users search for people on social networking sites by specifying their interests for example, cooking, in the search bar. The result shows a list of people who have mentioned cooking either in their interest section or somewhere in their profile name. But there is no information about their (users) level of interest in cooking. Object-centered sociality on the web emphasizes on linking people dynamically through shared objects of interests like jobs, workplaces, sports and hobbies. This technique involves analyzing content posted by a user on his/her homepage. Content analysis will provide the system with information about user’s interest level in various fields. When users post blogs, upload files or pictures they can annotate them via tags. Similar annotations are then searched through the system that determines the interests of other users. The result is further analyzed to determine the interest level of those users in that particular subject and create a compatibility meter. A friend list is then prepared and suggested to the user.

2 Background

The possibility to publish and gather personal information has been a major factor in the success of the Web from the beginning. Remarkably it was only in the year 2003 that the Web has become an active space of socialization for the majority of users [3]. That year has seen the rapid emergence of a new breed of Web sites, collectively referred to as social networking services (SNS). The first-mover Friendster attracted over 5 million registered users in the span of a few months [4] which was followed by Google and Microsoft starting or announcing similar services. Many popular Web applications are now exploiting user-driven information networks built by means of social annotations[5][6]. Social annotations are freely established associations between Web resources and metadata (keywords, categories, ratings) performed by a community of Web users with little or no central coordination. A mechanism of this kind, which has swiftly become well established is that of collaborative tagging[7][8], whereby Web users associate free-form keywords – called “tags” – with on-line content such as Web pages, digital photographs, bibliographic references and other media. The product of the users’ tagging activity is an open-ended information network – commonly referred to as “folksonomy” – which can be used for navigation and recommendation of content, and has been the object of many recent investigations across different disciplines[9][10][11].

Most Social Networking Sites focus on helping users create as many friend connections as possible, but provide limited
means of sustaining connections or keeping the friends interested in each other. Users of socializing sites typically get friend invitations also from authors they don’t necessarily know, or know very little about. These often include bands, communities or people wanting to network because of various reasons. All this is an effective way to encourage people to socialize more and to get connections, but the profile data in the sites needs to be discovered either manually or by proactive recommendation systems discussed e.g. in[12][13].

Several social networking, social bookmarking and blogging websites are being actively researched. While these SNSs allow searching for people with similar interest and make a friend connection, they do not provide any mechanism to determine a user’s interest level in a particular activity or subject. All past research acknowledges the challenge in connecting people on the web and increasing their network of friends, and sought similar solutions by suggesting them mutual friends.

2.1 The Future of Social Networks

The future of social networks on the Internet [2] suggests semantic web for linking and performing operations on object-related data gathered from different social networking websites. Friend-of-a-Friend8 project lets people share and use information about their activities and transfer information between websites. It describes people, the links between them and the things they create and do. Leveraging Social Networking Services to Encourage Interaction in Public Spaces[6] propose a system that utilizes the user generated content on socializing websites and displaying flashes of uploaded digital content on a large public display situated in the public or semi-public space like an airport or café. They do not, however, propose any methodology for making friends with similar interest level in an activity as that of the user’s. The success of any SNS is directly proportional to how they can keep their users interested. [15] suggests that tags that are used to organize resources or find related resources can now potentially serve as objects around which the users can form tight connections. Research in recent years have proposed developing an object-oriented sociality on the web that links people based on shared objects of interests like jobs, sports, hobbies etc., and therefore keep the users interested.

3 Architecture

3.1 System Architecture Overview

The system consists of the following components: User Management, Blog Management, Picture Management, Interests Management, Friend management, Friend Suggestion.

User Management – Facilitates adding users to the system authenticating users.

Blog Management – Facilitates a user to retrieve, upload or modify blogs in the system. To add a new blog, user clicks on the 'Add New Blog' link, adds the desired text and a title for the blog. The blog is then stored in the system. The user can later modify Blogs.

Picture Management – Facilitates a user to retrieve, upload or modify pictures in the system. To upload picture, user clicks on 'Upload Picture link. Picture can be uploaded from the desired location on the computer. A copy of the picture is then stored in the system.

Interests Management - Retrieves all interest in the system and display them in a tag cloud based on its usage. Searching an interest would display all blogs, files pertaining to that interest. It will also display interested users in a cloud form. A larger font size would mean more results were found for that user.

Friend Management – Facilitates a user to add a friend.

Friend Suggestion - The system matches a particular user’s interests with other users in the system and suggests friends to user. Users are displayed in cloud form. A larger font size would mean more similarity between the user and the suggested friend.

3.2 Friend Suggestion Algorithm

The implementation of the friend suggestion algorithm is a key component of this research. The following describes the major steps of the algorithm for building the Friend Suggestion List.

- Step 1. GetListOfUsersFromDB
  Get all the Users from the database that has the same tag names as the current user.

- Step 2. BuildTagInfoFromDB
  For a list of users that have the same tag names as the current user, extract all the tag names and tag count from the database and build the List<TagInfo> data structure.

- Step 3. ComputeUserMatrix
  Compute the User Matrix from the List<TagInfo>

- Step 4. NormalizeUserMatrix
  Compute the Normalized Matrix from the User Matrix

- Step 5. ComputeUserSimilarity
  Compute User Similarity from NormalizedUserMatrix

- Step 6. BuildFriendSuggestionList
  Build Friend Suggestion List from User Similarity in the previous step: List<FriendSuggester>:= List<userName, Similarity, Rank>
  Sort FriendSuggester by user Name.
4 Implementation

The implementation of our algorithms together with the presentation layer is called MyBook. A Spring framework based on MVC architecture is used to build this application. The following are the main pages provided by the presentation layer of the application:

Login Page: This page allows the users to enter Username and Password.

Home Page: This page shows the users’ friends, blogs, files and popular tags. It also lets the users navigate to other pages like AddBlog, UploadPicture, etc.

All the above modules are designed and developed using Java, JSP, Apache Tomcat, Spring Framework, and other open source technologies in the J2EE realm.

5 Experiments

Two experiments were carried out to illustrate the effectiveness of our design and implementation as follows.

5.1 Experiment 1

As shown in Figure 1, the user logins and enters blogs and tags them. The system looks into the user’s interests and suggests potential friends. As soon as the user enters more blogs and tags, the system updates the friend suggestion list.

Figure 1 – Friend Suggestion

5.2 Experiment 2

Following are the steps for step Experiment2. In this experiment, we create N users and generate M Tags. For each user, we create blogs with tags. Then we run the Friend Suggestion Algorithm.

1. Create N Users for the MyBook (u1..uN)
2. Generate M tags (t1..tM)
3. For each user, create \([M/2 + \text{Random}(0,M/2)]\) blogs with

Each blog a tag in \([\text{Random}(1, M)]\)

4. For each of the user, run the Friend Suggestion Algorithm and calculate the average response time in milliseconds.

In the following, Table 1 shows the time taken in generating the friend suggestion list in different scenarios such as the number of Users and number of Tags.

<table>
<thead>
<tr>
<th>Users</th>
<th>Tags</th>
<th>Generate the Friend Suggestion List Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>37.39</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>115.59</td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td>233.81</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>103.34</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>506.76</td>
</tr>
<tr>
<td>1000</td>
<td>50</td>
<td>1001.19</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>185.37</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
<td>954.41</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>1934.12</td>
</tr>
</tbody>
</table>

6 Conclusion

The system proved 100% efficient in finding a user’s interest and suggesting friends based on his/her interest during each browser session. It also updates the friend suggestion list as soon as any new blog or file is entered into the system. A manual search for friends with similar interests is not required. Friendships established by manual search may or may not lead to meaningful connection since the interest level of users in an activity cannot be determined beforehand. The system is efficient in automatically suggesting friends to a user based on his/her interest. In comparison to MyBook, most social networking sites suggest friends based on friend of a friend system. Facebook, MySpace and Orkut suggest mutual friendship. They also allow searching for people based on their interest but the results shown do not tell the interest level in that particular activity. It is therefore, not a viable solution to the issue addressed in this research. Delicious provides tagging and searching for data under the same tag name. It gives usernames who created that tag, but does not have profile information. It cannot be used as a utility to connect with people and socialize.
7 Future Work

As an extension to this work, we propose the scheme of auto-generation of Tags/Objects. The current implementation relies on user-entered tags to identify interest. In fact, Keyword search can be performed on the user-generated content (blogs, files etc) to auto-generate tags/objects. That can then be fed into the friend suggestion algorithm. Keyword search engine can either be run periodically on the user-generated content (blogs, files etc) to generate tags or can be performed immediately at the time the user-content gets saved. Implementation will require use of stand-alone text search engines like Lucene or full-text search operations provided by database engine (MySQL).

8 References.


