1. Abstract

In today's world of extensive social networking, individual recommender systems seem out of pace with the requirements of the day-to-day user's social life. A recommendation system, which makes suggestions to a group of users instead of individuals, has become increasingly important in both the workspace and people's social activities, such as attending social events together. Having said that, the leading social networking sites do not provide a group based recommender system. We propose an event based group recommendation method that utilizes the social interest of group members. We study the key interests of individuals to deduce commonalities among them to form groups and propose upcoming music events and upcoming currently showing movies. A study of diverse groups demonstrates the effectiveness of proposed techniques and the importance of incorporating the social interest in group recommendation system.

2. Introduction

In general, individuals spend majority of their leisure time in searching for ways to spend quality time and then looking for available people who can join them. If this time and effort to search could be minimized, it would help maximize the time spent in the actual experience. There are websites available which are dedicated to upcoming events' listing. To add to this there is now an increased availability of personalized information, thanks to social networking websites like Facebook and Twitter. Both these sets of information if brought together make a good case for enhancing the user experience of social activities and networking. This ensures saving the time and effort of going through hundreds of events by suggesting only relevant events and friends with similar interests. Thus, the main purpose of this recommender system is to work over group based dynamics and come up with recommendations common to members of the group.

3. Description

After a brief skimming through social networking sites, it was found that Facebook had the most active users and therefore the most information about
people and their interests. Facebook is therefore used as a data source to obtain personal information about people when authorized by them. This implementation of the group based recommender system is thus in the form of a Facebook App.

3.1 Data Collection and Cleaning

The system is designed to suggest musical events or now showing movies that a group of people could go to together. The next task therefore is to find data sources that list upcoming musical event and/or listings of recently released and upcoming movies and their schedules.

3.1.1. Gathering data for Movies

Data source for finding movie information is IMDB. This data was extracted from 1) http://www.imdb.com/movies-coming-soon/
We obtained data worth 100 distinct movies per city which amounts to approximately 1100 records considering theatres and show times over a period of 2 weeks.

Once this information is obtained, it is loaded in Karma. Relationships between different properties are defined using ontologies like foaf, time, schema, geonames and dcmitype. This ontology is defined as part of local Uri http://events.edu/movies which are then converted to a .n3 representation using Karma and are loaded into the repository on the fusion data server as a triple store.
3.1.2. Gathering data for Music

Data source for finding music information is www.songkick.com. This data was extracted from http://www.songkick.com/metro_areas/17835-us-los-angeles using Easy Web Extract and cleaned using Google Refine. We obtained data worth 1000 music events per city which amounts to approximately 1400 records considering venues and show times over 2 weeks. Procedure to represent data in Karma is same as described for Movies.

3.1.3 Gathering data from Facebook

Once the user authorizes the access to their likes and that of their friends, we first extract the current users like for music and movies, then we extract the friend list of the current user and their likes for music and movies using Facebook JavaScript SDK.

3.1.4 Notable Issues

- The list of likes of user’s and friends retrieved from Facebook are simply a list of Facebook URLs which make it hard to understand whether the URL liked is that of a person, a band or a movie.
- Dates extracted from websites have to then be resolved into Day, Date, Month, Year, Hour so as to map them to the TIME ontology as per W3C specification.
- The likes of users obtained can be an artist in a band, or a band or an actor or a movie. Record Linkage of the artist to the band and then to the event in which they are performing is another major challenge in this process. Handling of these issues is explained in Section 5.

4. Architecture

As shown in figure 1, data once scraped from IMDB and Songkick is stored in a triple store of RDF events. User’s information from
Facebook is sent to the recommender engine which then looks for relevant events in the triple store based on those likes. Groups are then formed for the shortlisted events and friends having similar interests. These events and respective groups are then suggested to the user.

5. Workflow

5.1 Scenario 1 - Current Events

Current Events show that the events which are already RSVP’d by the user. We maintain a separate repository for such RSVP’d events.

5.2 Scenario 2 - New Events

New Events show the upcoming events that are recommended to the group of friends based on similarity of interests. Workflow is as follows.

- Get the current user information and the friend list.
- Filter the friend list based on the location of the current user using location IDs provided by Facebook. This handles Entity Extraction for user’s location. Get the movies and music likes for friends.
- For each like of current user, get the relevant events from the triple store using SPARQL queries.
- The SPARQL query for each like of music looks up first in the list of artists that are performing at the event. If no match found, it is assumed that the like could be a band, and its name is looked up in the list of Bands or Event names in the triple store. If still no match, it means there is no event relevant to that like of music by the user and we move on to the next like in the list. This is how record linkage is handled between artists and their bands, bands and the events they are performing at, artists and the movies they are part of.
- For every event thus obtained, get all friends who have similar likes.
- Likes of friends are compared with event details using Jaro-Winkler String Comparison. Threshold used for Jaro-Winkler String Comparison is 0.90.
- Each event is then recommended to the main user along with a list of friends of who might be interested in going to that event.
6. Conclusion

It was observed that if a certain event had multiple artists performing and were part of interests of different friends, it would appear as a recommendation to the main user with all relevant friends displayed as part of the group. This method of implementation thus results in the system finding more events that could be related to a user.

- How well does it work?
  2-3 out of 4 times, if available, relevant events based on artist likes are displayed. Common friends with similar likes (different artists of the same band) are displayed as part of a group.

- How did we measure it?
  Events displayed for bands of artists liked were compared with the ones in the triple store which accounts for the precision of the results.

7. Future work

More information related to the movies/music can be fetched from DBpedia. The ability of the system can be enhanced if relation between artists based on genre can be established to suggest the events based on a person’s interest in a type of movie or music than his/her interest in a particular artist. The sustainability of the application can be maintained by automating the data scraping functionality for keeping the data updated.

8. References