EasyGo: An Event-based Social Network

Yinuo Zhang, Hao Wu

Department of Computer Science
University of Southern California
Los Angeles, CA 90089
{yinuozha, hwu732}@usc.edu

Abstract. People attend social events to share hobbies, enjoy networking experience, and build long lasting friendships. However, one may find it difficult to keep track of upcoming events around and look for someone who has common interests to join an event together. In this paper, we present EasyGo, a web-based mashup application which integrates information of events, tickets and group purchasing from various online sources. This would help promote group events and match people with common interest to go for events together.

1 Introduction

Nowadays, people enjoy various information of social events online such as concerts, sports, theatres and so on. Those social events, especially group events provide the opportunity for people to share hobbies and social networking. However, it is not easy for one to keep track of upcoming events as well as to find people who have common interests and are interested in join an same event together. In this paper, we introduce EasyGo, a web social networking application which integrates information of events, tickets and group purchase from various online sources including StubHub\(^1\), Barry’s Tickets\(^2\) and Vivid Seats\(^3\). Users can search events on EasyGo by keyword-based query. We also adopt recommendation algorithms to personalize event information displayed on the front page of each user. To improve users’ social experience and reduce fees of ticket purchase, our web system allows users to create and join a group to purchase group tickets and join an event together.

Users can gain various benefits by using Easygo: 1) easy to discover the deal with lowest price among multiple online ticket markets; 2) save money by joining a group to share the service and delivery fee; 3) make friends and social networking with other users who share similar interests and attend events together.

\(^1\) http://www.stubhub.com/
\(^2\) http://www.barrystickets.com/
\(^3\) http://www.vividseats.com/
2 EasyGo Application

We develop EasyGo with ASP.NET and deploy it using web server IIS[^4]. EasyGo can be accessed through the web link: [http://huanghe.usc.edu/easygo/](http://huanghe.usc.edu/easygo/)

2.1 Registration and Login

![Image](image1.png)

Fig. 1. Registration in EasyGo: (a) Tradition Registration. (b) Facebook Registration.

EasyGo provides event-based social networking services. Registration is required for new users. There are two ways to sign up. One is to specify personal information including user name, email and password as registration information (Figure 1(a)). The other is to connect to a user’s Facebook account for a fast registration (Figure 1(b)). The latter is preferred since the user Facebook profile can be used to identify personal interests for event recommendation. After login, a user may connect to their friends, specify personal interests, search events and create, join a group for ticket purchase.

2.2 Recommendation and Search

The home page of EasyGo after a user logs in is shown in Figure 2(a). Based on personal interests the users have specified in “My Profile”, our system will display relevant upcoming events that the user may be interested as recommendations on the front page.

A user can query future events by typing in keywords in the search box. A list of relevant events will be retrieved and displayed in chronological order (Figure 2(b)). Our system also enables Google Maps view of the event location by choosing “Map” option (Figure 2(c)). The user can zoom in the map and click on the link of a specific event (Figure 2(d)).

[^4]: [http://www.iis.net/](http://www.iis.net/)
Fig. 2. The front-end of EasyGo: (a) Interface for event search and recommendations. (b) A list of searched events. (c) A map of event locations. (d) Selected event on the map

2.3 Deals and Group Purchase

Once a specific event is clicked, the user is directed to the event page (Figure 3). A table shows all deals on which one may create a group. The deal information including the total number of tickets, the price per ticket and the ticket source. A user can create a group on each deal by clicking the “Create” button. If some groups have been formed for this event, our system also shows the group information includes the total number of tickets for this deal, the price per ticket excluding service charge and delivery fee, the ticket source, the creator of this group, the spaces still available and the saved amount when joining this group. The user hence can join the group by clicking the join button in the same row. Email notification is also enabled in our system if a user perform group joining and initializing. A user can also view their friends’ activity to follow a group that they have already joined.
3 System and Methodology

3.1 Triple Store Generation

The key component of our system is the triple store which stores the ontology information including events and users for retrieval. We build the event ontology (Figure 4(a)) by crawling information from popular online ticket trading platforms including StubHub, Barry’s Tickets and Vivid Seats. To extract event information from each website, we either use individual APIs from the websites or write a wrapper manually using Scrapy. The specific fields of an event we extract include name, venue, city, state, time and web link. After collecting the events from different websites, we perform a data integration task on the events. Since different websites have different representations for the events, it is not a trivial problem. We use some record linkage techniques to link the information referring to the same event. Finally, we generate the ontology representation for each event using Karma. Moreover, we also use open linked data to enrich the event ontology. DBpedia and Linked GeoData are used to expand the spatial information of events.

However, our ontology repository does not maintain the deal information of an event due to the consideration of efficiency and the dynamic property of deals. We retrieve the information of deals on the fly in the query execution step (driven by user search or event link click) instead of being stored in our

5 http://scrapy.org/
6 http://dbpedia.org/
7 http://linkedgeodata.org/
triple store. For detailed deal information extraction for each event, we use API provided by each website (e.g., StubHub API).

Figure 4(b) illustrates how the user ontology is generated in a typical scenario. For example, Kevin is a registered user in our system. The user triples are generated during the registration phase. He may search using keywords (i.e., “Lakers vs Heat”) for events of interests to him, or reads upcoming events from our recommendation list. Once an event is clicked, our system automatically retrieves the deals information from different data sources including StubHub, Barry’s Tickets and Vivid Seats. At the same time, the group information for that event is also retrieved from the group ontology. Kevin can either initialize a new group on a deal or join a group with available spaces. If Kevin creates a new group, the triple corresponding to this group will be generated in the group ontology. If Kevin chooses to join an active group (i.e., Group 1 in Figure 4(b)), the triples corresponding to the friendship between Kevin and other group members will be generated. The creator will be directed to the payment page on the online marketplace. After payment, all group members can go together to enjoy the game.

3.2 Recommendation Capability

We also enable recommendation capability in our system to personalize event information on the front page. Specifically, we adopted several machine learning approaches proposed in our previous work. In the early usage of our system, we mainly use Similarity Based Approach (SBA). We use Latent Dirichlet Allocation (LDA) to extract the topic distribution over each user and each event, and calculate the similarity between their distributions. The recommended events are mainly selected from the events with high similarity with a user. Each user’s information for generating topic distribution is either extracted from explicit profile user inputs in the registration phase or implicit profile from Facebook. Event profiles come from the event ontology. For the long run, once our system is

\[\text{http://www.stubhub.com/ticketAPI/restSvc/event/event_id}\]
used by more and more people, we will adopt other approaches like Relationship Based Approach (RBA), History Based Approach (HBA) and Hybrid Approach (SRH) which take friendship and history into consideration. More details about above approaches can be found in [1].

4 Conclusions

In this paper, we present EasyGo, a web-based social network integrating multiple online data sources for event search, group purchase of event tickets and social networking. EasyGo personalizes events recommendation based on individual interests of each user. For the future direction, we would like to explore the dynamism in coming events. Currently, we manually set a interval (e.g., one day) to refresh the event ontology in order to add newly emerging events into our system. We plan to propose a machine learning based framework which can automatically adjust the frequency based on past event records.

5 Acknowledgements

We would like to thank Prof. Pedro Szekely, Prof. Jose Luis Ambite and George Konstantinidis for helpful suggestions and guidance on this work.

References