Integration of Heterogeneous Web Services for Event-based Social Networks

Yinuo Zhang Hao Wu Anand Panangadan
Viktor K. Prasanna
University of Southern California
Los Angeles, CA 90089, USA
{yinuozha, hwu732, anandvp, prasanna}@usc.edu

Abstract

Event-based online social networks are Internet-based services that enable users to participate in real world experiences together. Event-based social networks can be created by a community of end-users based on their own interests in specific types of event and sources of event information. We propose a method to create such event-based social networks through integration of existing online information sources of events using a Semantic Web framework. In order to match people with common interests in such activities to self-organize into a social network, we integrate information from heterogeneous information sources related to event schedules, ticket purchases, and group attendance from multiple online sources. The Semantic Web framework is used to represent these heterogeneous datasets and unstructured online data is converted into ontologies. Links between event information in different sources are discovered using both the syntactic similarity and semantic similarity between ontology classes. We use an approach based on Latent Dirichlet Allocation (LDA) over the space of topics related to each event and user profiles for event recommendation. This enables the event-based social network to recommend friends based on shared interest in an event — online friendship is established after mutual attendance of the same event. We demonstrate this approach with EasyGo, a web-based mashup application which integrates information of events such as concerts, sports, theatres, as well as tickets and group purchase from multiple online sources.

1 Introduction

Event-based online social networks are Internet-based services that enable users to participate in real world experiences together. With the increasing availability of location-based information and services, there has been a corresponding increase in the variety of event-based online social services. For instance, Facebook has realized the importance of social events and "Events" is now listed as one of four basic application types on their front page (the others being News Feed, Messages, and Photos). However, such event-based networks are typically created in a "top-down" approach in that the complete online system is designed with a specific framework for representing events and for subscribing end-users to these events. On the other hand, we envision an approach where event-based social networks can be created by a community of end-users based on their own interests in specific types of event and sources of event information.

In this work, we propose a method to create such event-based social networks through integration of existing online information sources of events using a Semantic Web framework. We focus on activities which enable participants to share a real-world experience such as attending a concert, theater, or sports event. In order to match people with common interests in such activities to self-organize into a social network, we integrate information from heterogeneous information sources related to event schedules, ticket purchases, and group attendance from multiple online sources. Specifically, instead of retrieving the events and tickets information from one source (such as Plancast[1] and Yahoo! Upcoming[2]), our approach integrates information from multiple sources (for instances, ticket prices for the same event from different sources such as StubHub[3], Barry's Tickets[4], and Ticketmaster[5]).

Challenges: Building such a system requires addressing three main technical challenges. First, information from heterogeneous sources have to be represented in a uniform manner that enables the system to reason over the available choices and select the most appropriate subset of the information that is relevant to a user. We propose to use the Semantic Web framework to provide such a representation.

[1] http://www.plancast.com/
[2] http://upcoming.yahoo.com/
[3] http://www.stubhub.com/
[4] http://www.barrystickets.com/
[5] http://www.ticketmaster.com/

Since few existing sources present their information in Semantic Web standards, we present a method for converting unstructured data collected from the Web into structured information, specifically Ontologies. Second, links between event information in different sources have to be discovered, i.e., semantic entities representing the same physical entities have to be identified. In our approach, we use both the syntactic similarity and semantic similarity between ontology classes to discover such links. The linkages are represented in the ontology which enables the system to use Semantic Web standards to make use of the discovered relationships between the data sources. Third, an *event recommendation* solution has to be included in the system. Event recommendation in our context refers to suggesting to a user other users who may be interested in the same event, and other events which may be of interest. Most online social networks (including Facebook) recommend friends based on the number of mutual friends of a user's friends. An event-based social network opens the possibility of friend recommendation based on shared interest in an event. Online friendship is established after mutual attendance of the same event. We use an similarity-based approach based on Latent Dirichlet Allocation (LDA) over the space of topics related to each event.

We illustrate our approach of building an event-based social network from existing online sources with *EasyGo*, a web application (mashup) which integrates event content from multiple websites and recommends events and fellow attendees to a user. A preliminary demonstration of *EasyGo* was presented in [13]. In this paper, we give a more detailed description and analysis of the system. In particular, we describe the ontology that forms the framework for integrating heterogeneous information sources and the knowledge base for the system, the approach based on topic modeling for event recommendation, and the user interface.

The contributions of this paper are as follows.

1. a Semantic Web-based approach that integrates heterogeneous online data sources related to events
2. an approach that makes use of syntactic and semantic similarities between ontologies to discover linkages between heterogeneous sources
3. an approach based on LDA for event recommendation, and
4. demonstration of the approach using a web-based application.

The rest of the article is organized as follows. Section 2 gives an overview of the related work. Section 3 presents the system architecture. Section 4 describes event recommendation for building an event-based social network. Section 5 introduces a web-based application, EasyGo, which is built based on the proposed approach. We conclude in Section 6.

2 Related work

2.1 Event-based Social Networks

Event-based social networks (EBSNs) enable both online social interactions and offline social interactions among users. It is formally defined as a heterogeneous network $G = \{U, A^{on}, A^{off}\}$, where U represents the set of users (nodes), A^{on} corresponds to the set of online social interactions (edges), and A^{off} represents the set of offline social interactions (edges) [4].

Meetup[6] is a popular event-based social networking service which has already been attracting research interests on large scale online and offline social data analysis. [8] investigated the social behaviors of users participating events on Meetup and concluded that they share similar social structures. Offline social activities are also associated with checkin actions. Such checkins can indicate social interactions to some degree [1, 5]. At the same time, the location information can be used to infer social ties [9].

2.2 Recommendation in Social Networks

In social networking services, a recommendation system automatically identifies information relevant for a given user. This capability is used for influence analysis and targeted online marketing. [6] presented a model which combines geographical information of users and content of items to achieve a better rating perdition. [11] considered users' co-tagging behaviors and added the similarity relationship to the graph to improve the recommendation performance. [2] proposed a social data integration framework which can facilitate the prediction of health conditions.

Event recommendation plays an important role in establishing social connections. In our previous work [14], we proposed a recommendation system based on the similarity of an event and a user in terms of topics. We also considered social ties and attendance history to increase recommendation accuracy. [12] proposed a new group recommendation method based on event-based social networks. [7] presented a Baysian probability model which considers the heterogeneous social impact and implicit feedback characteristic in the event recommendation.

3 System Framework

The system architecture is shown in Fig.1. A Triple Store (TS) provides the framework for event information integration. All information related to events and users is stored in TS in the form of an ontology. In order to construct TS, we

[6]http://www.meetup.com/

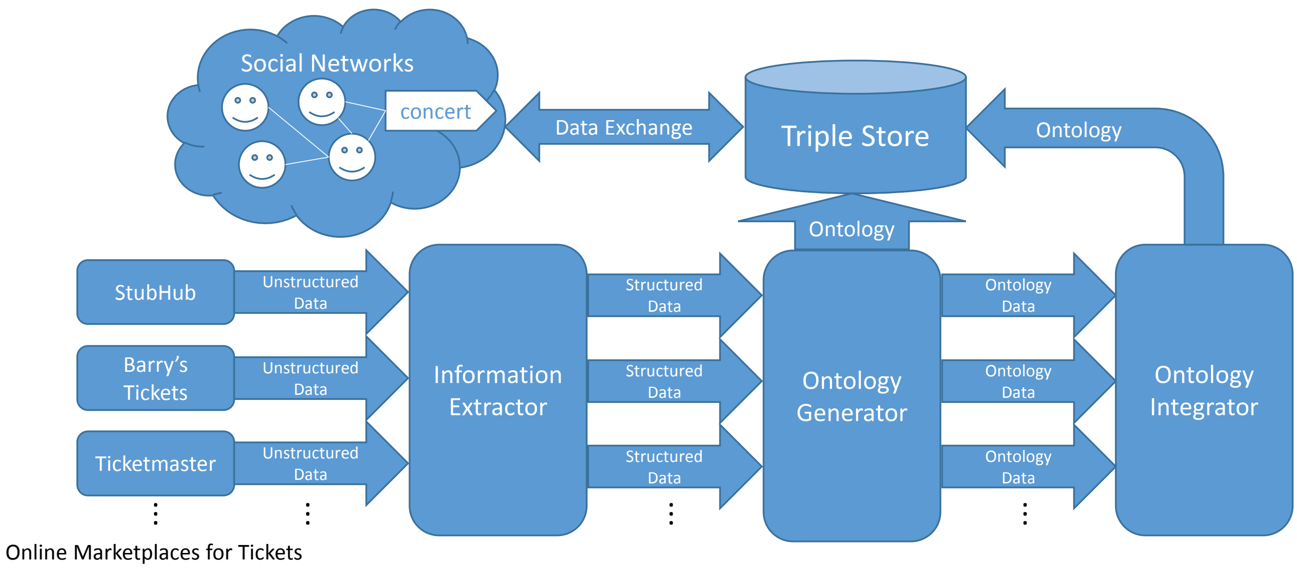

Figure 1. System Architecture

build three components in the system: Information Extractor (IE), Ontology Generator (OG), and Ontology Integrator (OI).

IE converts unstructured raw data gathered on the web to structured information. This information covers basic aspects of events such as name, time, venue, and performers. OG takes this information as input and generates the semantic representation for the events in each source. In order to accommodate for the heterogeneity existing in the event information, OI discovers the alignments between events in different sources. Such alignments are created in the form of ontology and stored in TS. TS not only provides the knowledge base for the EBSN but also carries the semantic representation of the EBSN. We describe these three components in greater detail below.

3.1 Information Extractor

Data Sources

In order to acquire the information of various upcoming events, we consider several online ticket trading marketplaces as our event sources. Such marketplaces include:

- **StubHub** is an online marketplace owned by eBay, which provides services for buyers and sellers of tickets for sports, concerts, theater, and other live entertainment events.
- **Barry's Tickets** is an online ticket provider for all sports, concert, theater, and exclusive event tickets. They also provide tickets to special events that no other website offers.
- **Ticketmaster** is an online ticket selling website for various sports, concert, and theater events. In contrast to the previous two websites, tickets on Ticketmaster are usually listed officially by organizers of the events.

Heterogeneity exists in these marketplaces. For example, StubHub has the largest number of tickets and, usually, the cheapest ones. Barry's Tickets is the largest in terms of the number of events since it has some special events. Ticketmaster has the most tickets for those events which are just published. Moreover, the ticket price is fixed at the official price set by event organizers. Such heterogeneity is one important reason why we integrate information from different marketplaces. Our system is capable of supporting more than three platforms.

Information Extraction: We use Scrapy[7] for the web crawling task. We recursively crawl the basic information of events such as name, location, time, and purchase link. We output this information in a .json file. For example, an event named "Carrie Underwood" discovered on StubHub is represented as,

```
{"City": "Abbotsford",
"Name": "Carrie Underwood",
"Venue": "Abbotsford Entertainment and
Sports Centre",
"State": "BC",
"API_Link": "www.stubhub.com/.../4188319/",
"Link": "www.stubhub.com/...4188319/",
"Time": "Thu, 05/23/2013 7:30 p.m.
PDT"}
```

API_Link has the link to all deals of that event (for the

[7] http://scrapy.org/

system to retrieve deal information) and *Link* has the link to the webpage of that event (for users to purchase the tickets). For efficiency, we only maintain basic event information. The information of deals is retrieved real-time in the query execution phase instead of being stored in the triple store. Once IE generates this structured information, OG takes it as input for the ontology generation process.

3.2 Ontology Generator

In order to generate the semantic representation of events, we use Karma[8] Fig.2 for our ontology generation task. Karma is a semantic web tool that enables users to quickly and easily convert data from a variety of data sources including databases, spreadsheets, delimited text files, XML, JSON, KML and Web APIs to ontology in the form of RDF. In RDF, each piece of knowledge is represented as a triple. For example, the StubHub event ontology has 85,392 triples. After we generate the ontologies for all event sources, we store them in TS. We use OpenLink Virtuoso[9] as our triple store server. OpenLink Virtuoso is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, ORDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system.

Figure 2. Using Karma to generate the semantic representation

3.3 Ontology Integrator

The information in TS generated from different sources is still isolated at this stage. The goal of OI is to match the information from heterogeneous event sources belonging to the same event. Besides the coverage of events and tickets, different marketplaces also have different naming conventions and information representations. For example, “location” and “time” (StubHub) are named as “venue” and “date” in Barry’s Tickets. To solve this problem, we consider both *syntactic similarity* and *semantic similarity*.

For syntactic similarity, Levenshtein distance[3] is used as the distance metric between the names of the two properties. Formally, the syntactic similarity, $s_{sy}(E_1, E_2)$ between two properties E_1 and E_2 is

$$s_{sy}(E_1, E_2) = 1 - \frac{Lev(E_1, E_2)}{max(|E_1|, |E_2|)} \tag{1}$$

where E_i denotes the name of ith property, $|E_i|$ is the length of the name string, and $Lev(w_1, w_2)$ is the Levenshtein distance between two words w_1 and w_2.

For semantic similarity, we first tokenize the names of both properties. Denote by $E_i.\text{TOK}_j$ as the j-th token in the name of property E_i. Then we retrieve the synset of each token using Open Linked Data (WordNet)[10]. Denote by $syn(w)$ the WordNet synset of a word w, and calculate the Jaccard similarity[11] on the synsets of each pair of tokens. Finally, we return the average-max Jaccard similarity as the semantic similarity:

$$s_{se}(E_1, E_2) = \frac{1}{n} \times$$

$$\sum_i \max_j Jac(syn(E_1.\text{TOK}_i), (syn(E_2.\text{TOK}_j)))$$

Here, $Jac()$ represents the Jaccard similarity between two sets and n is the number of tokens in E_1’s name.

The final similarity is a weighted sum of the syntactic and semantic similarities:

$$s(E_1, E_2) = \omega_{se} s_{se}(E_1, E_2) + \omega_{sy} s_{sy}(E_1, E_2) \tag{2}$$

The weights are pre-defined as system parameters. In this work, $\omega_{se} = \omega_{sy} = 0.5$.

For example, consider two properties named **venue** and **location**. The syntactic similarity $s_{sy} = 1 - \frac{8}{8} = 0$ and the semantic similarity $s_{se} = 0.83$. So $s(venue, location) = 0.42$. If the similarity is larger than a threshold, say 0.3, then we consider those properties are matched.

Next, we consider instance-level matching which retrieves identical events. The following steps are used to discover the matches:

1. Consider a pair of events as a potential match only if they have identical location and time information.

2. For each pair of potential matched events (e_1, e_2), calculate the event similarity $s(e_1, e_2)$ as

[8] http://www.isi.edu/integration/karma/
[9] http://virtuoso.openlinksw.com/
[10] http://wordnet.princeton.edu/
[11] http://en.wikipedia.org/wiki/Jaccard_index

$$\omega_{Jac} Jac(TOK(e_1), TOK(e_2)) + \omega_{SWS} SWS(e_1, e_2)$$

where $Jac(TOK(e_1), TOK(e_2))$ calculates the Jaccard similarity between the token sets of the names of e_1 and e_2, and $SWS(e_1, e_2)$ calculates the Smith-Waterman Similarity [10] between the names of e_1 and e_2.

The Jaccard similarity eliminates the negative effect of word ordering. For example, consider an event named *"Taylor Swift and Ed Sheeran"* and another event named *"Ed Sheeran and Taylor Swift"*. The Jaccard similarity is 1 in this case. The Smith-Waterman Similarity identifies local sequence alignment. For example, *"Taylor Swift"* and *"Taylor Swift featuring Ed Sheeran"* has higher Smith-Waterman Similarity compared with Levenshtein Similarity. Similar to the property similarity, the weights are predefined as system parameters. In this work, both of them are 0.5. We also use open linked data to enrich the event ontology. DBpedia[12] and Linked GeoData[13] are used to expand the spatial information of events. Once we discover all matches, we store them in the form of triples in TS.

4 Event Recommendation for Event-based Social Networks

Figure 3 illustrates the process of building a social network from the information stored in TS. This is illustrated in the following example. Kevin is an existing user in the system. He searches for events which are interesting to him by typing keywords such as *"Lakers vs Heat"*, or selects one of the events in his personalized recommendation list. Once an event is identified, the system automatically retrieves ticket deals from different data sources such as StubHub and Barry's Tickets. For example, StubHub provides an API link in the format of *http://www.stubhub.com/ticketAPI/restSvc/event/event_id)* which has detailed information on each deal for a specific event. Kevin can either initialize a new group on a deal or join a group with a vacancy. The information of existing groups for an event is retrieved from TS. If Kevin creates a new group, the information related to this group such as creator, members, number of tickets and number of vacancies will be generated in TS. If Kevin chooses to join an existing group (e.g., Group 1 in Fig 3), *friendship* between Kevin and other group members is established and stored as triples in TS. At the same time, the triples corresponding to that group is also updated. Each member is then directed to a payment page on a specific online ticket website once the group is full. After all payments are settled, the members proceed to the event and potentially form a social circle in real life.

Social network formation requires an event recommendation capability. We adopt the machine learning approach proposed in our previous work [14], specifically *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 selected from the events with high similarity to a user. Each user's information for generating topic distribution is either extracted from the explicit profile of a user's inputs in the registration phase or implicit profile from an existing service such as Facebook. Event profiles are obtained from the event ontology. We adopt cosine similarity (Equation 3), although other approaches can be used. $S_1(u_i, e_j)$ is the recommendation score of event e_j for user u_i.

$$S_1(u_i, e_j) = cos(\overrightarrow{\theta}_{u_i}, \overrightarrow{\theta}_{e_j}) = \frac{\overrightarrow{\theta}_{u_i} \cdot \overrightarrow{\theta}_{e_j}}{||\overrightarrow{\theta}_{u_i}||||\overrightarrow{\theta}_{e_j}||}, \quad (3)$$

As future work, we will evaluate other methods such as *Relationship Based Approach (RBA), History Based Approach (HBA), and Hybrid Approach (SRH)* which take friendship and history into consideration once the population in our system is large enough. More details about these approaches are provided in [14].

5 Application

EasyGo is a web-based application developed based on the proposed Semantic Web framework. The front-end of EasyGo is shown in Figure 4(a). A user has two options to find interesting events (e.g., sports and concerts) He/she can either search for events by keywords or select an event in the recommendation list. For example, a user retrieves future events by providing keywords in the search box. After the keywords are received by the system, the relevant events stored in TS will be extracted. Specifically, the search engine explores the name, description, and location properties to find the matched events. Then, a list of events is displayed in chronological order (Figure 4(b)). The matched events can also be shown on a map by choosing "Ma" option (Figure 4(c)). We enabled Bing Maps API on EasyGo to give users a geospatial filter on the events.

By selecting a specific event, say an NBA game, the user is directed to the event page (Figure 4(d)) where two tables are displayed. The first one shows the group information for this event which is generated based on the triples in TS. The 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 the group, the

[12]http://dbpedia.org/

[13]http://linkedgeodata.org/

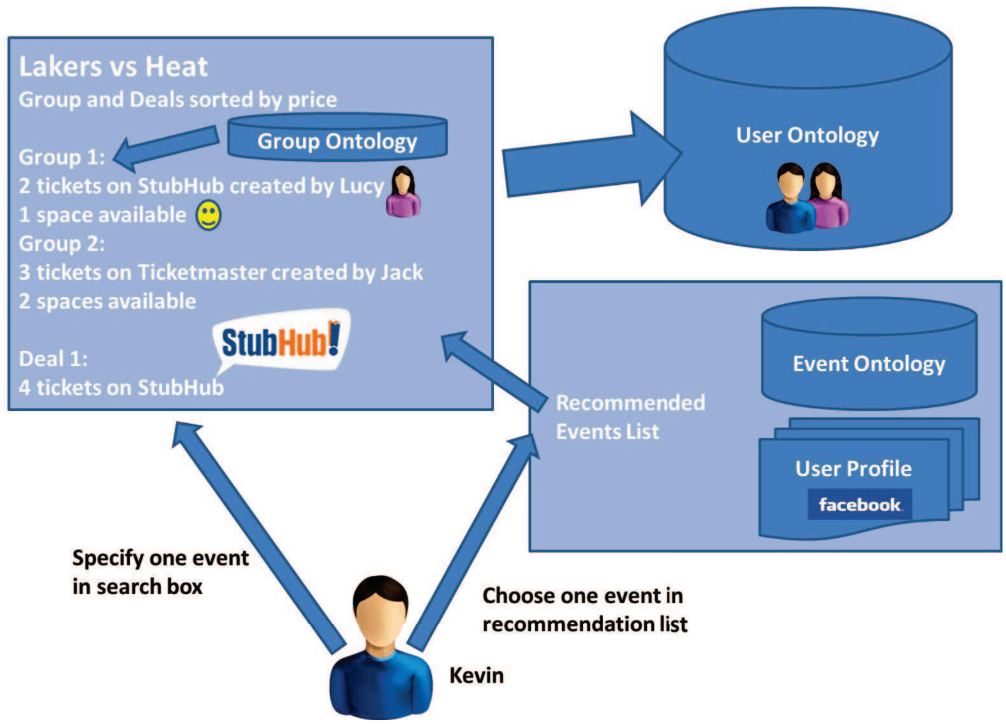

Figure 3. Attending an Event

number of vacancies, and the amount of money that can be saved when joining this group. Once a user joins a group by clicking the "Join" button, both the group ontology and user ontology in TS are updated. Specifically, the user information will be linked to the group in the group ontology and the friendship will be established among group members in the user ontology. At the same time, an email is sent to the creator. The second table shows all deals for which no group has been formed. The deals information is retrieved by accessing the event API link stored in the event ontology. EasyGo crawls all valid deals for that event through the link. All information, including the total number of tickets, the price per ticket, and the ticket source is shown in the table. A user can initialize a group on this deal by clicking the "Create" button. Similarly, the group ontology in TS will be updated based on the action.

In order to maintain user and friendship records, registration is required for this application. A user has two ways to sign up. One way is to input personal information such as user name, email and password as basic registration information (Figure 5(a)). An alternative is to connect to a user's Facebook account for fast registration (Figure 5(b)). The second choice is preferred since the user's Facebook profile can be used as the source for event recommendation. All user information is stored in the user ontology in TS.

6 Conclusions

We presented an approach to build an event-based social networking system from existing online information sources. The system extracts events from multiple online ticket trading platforms and generates an integrated event semantic knowledge base, specifically a triple store. The approach is demonstrated using the web-based application, EasyGo. EasyGo is built around the triple store and uses Latent Dirichlet Allocation to generate event recommendation based on user profiles. EasyGo provides the following capabilities. 1) a user can discover the deal with lowest price among multiple online ticket markets; 2) a user can further save money by joining a group to share the service and delivery fee; 3) a user can establish friendships with other users who share similar interests and attend the same events.

For future work, we intend to explore the dynamism in upcoming events. Currenly, 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 framework which can automatically adjust the update frequency based on past event records.

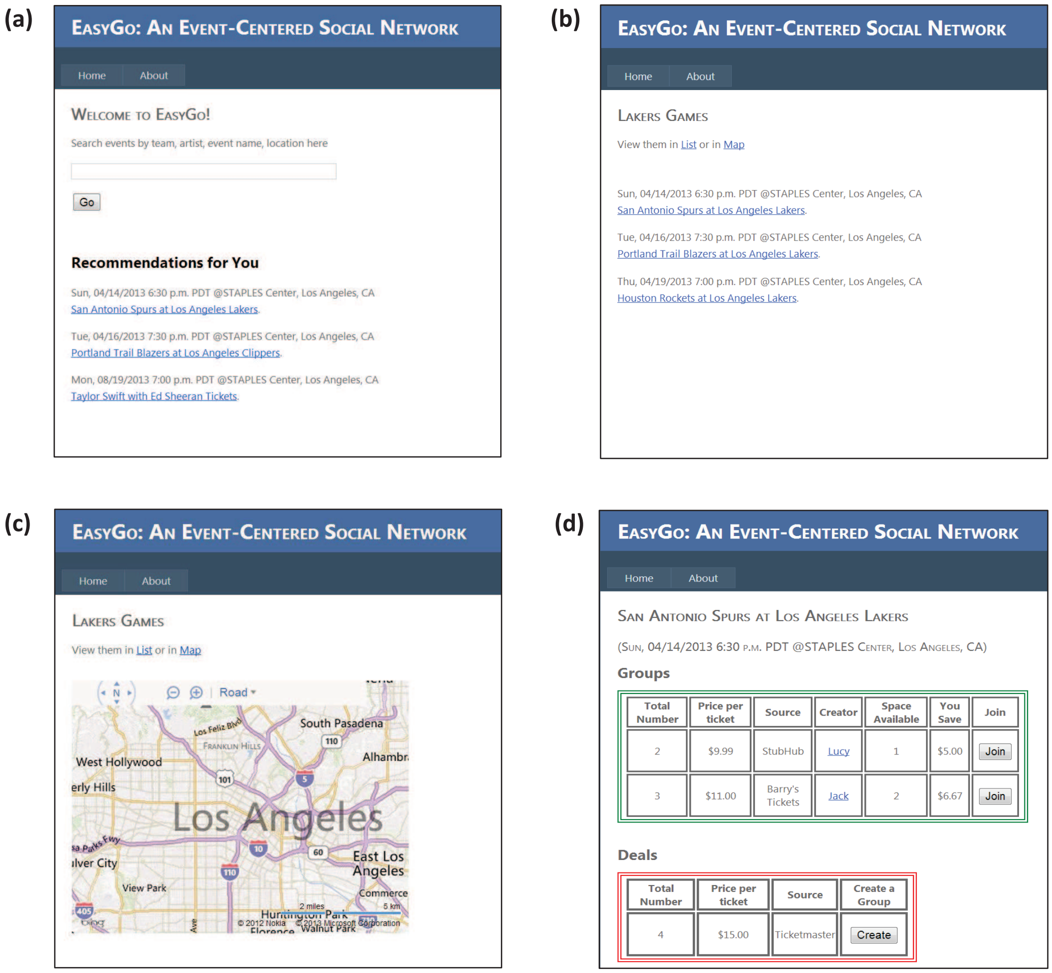

Figure 4. The front-end of EasyGo: (a) Interface for event search and recommendations. (b) A list of searched events. (c) A map of searched events. (d) Group and deal information for an event.

References

[1] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1082–1090, 2011.

[2] Xiang Ji. Social data integration and analytics for health intelligence. In *VLDB PhD Workshop*, 2014.

[3] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics-Doklady*, 10(8):707–710, 1966.

[4] Xingjie Liu, Qi He, Yuanyuan Tian, Wang-Chien Lee, John McPherson, and Jiawei Han. Event-based social networks: linking the online and offline social worlds. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1032–1040, 2012.

[5] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011.

[6] Zhi Qiao, Peng Zhang, Jing He, Yanan Cao, Chuan Zhou, and Li Guo. Combining geographical information of users and content of items for accurate rating prediction. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea,*

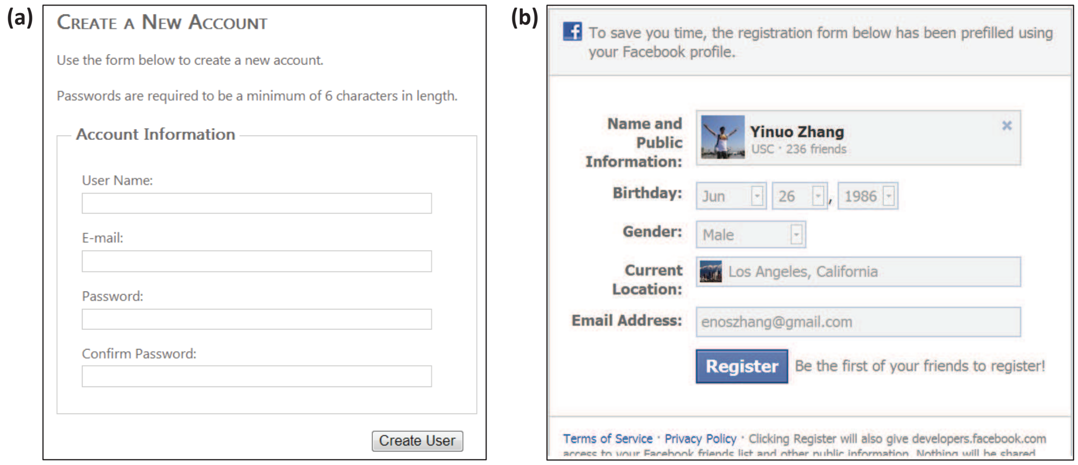

Figure 5. Registration in EasyGo: (a) Tradition Registration. (b) Facebook Registration.

April 7-11, 2014, Companion Volume, pages 361–362, 2014.

[7] Zhi Qiao, Peng Zhang, Chuan Zhou, Yanan Cao, Li Guo, and Yanchuan Zhang. Event recommendation in event-based social networks. In *28th AAAI Conference on Artificial Intelligence*, 2014.

[8] Thomas H. Sander and Saguaro Seminar. E-associations? using technology to connect citizens: The case of meetup.com. In *Annual Meeting of the American Political Science Association*, 2005.

[9] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1046–1054, 2011.

[10] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197, 1981.

[11] Ziqi Wang, Yuwei Tan, and Ming Zhang. Graph-based recommendation on social networks. In *Advances in Web Technologies and Applications, Proceedings of the 12th Asia-Pacific Web Conference, APWeb 2010, Busan, Korea, 6-8 April 2010*, pages 116–122, 2010.

[12] Wei Zhang, Jianyong Wang, and Wei Feng. Combining latent factor model with location features for event-based group recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 910–918, New York, NY, USA, 2013. ACM.

[13] Yinuo Zhang and Hao Wu. EasyGo: An event-centered social network. In *AI Mashup Challenge, Extended Semantic Web Conference (ESWC)*, 2013.

[14] Yinuo Zhang, Hao Wu, Vikram Sorathia, and Viktor K. Prasanna. Event recommendation in social networks with linked data enablement. In *ICEIS*, 2013.