A Cross-Service Travel Engine for Trip Planning

Gang Chen #1, Chen Liu †2, Meiyu Lu †3, Beng Chin Ooi †4, Shanshan Ying †5
Anthony K. H. Tung †6, Dongxiang Zhang †7, Meihui Zhang †8
#1College of Computer Science, Zhejiang University
#1cg@cs.zju.edu.cn
†School of Computing, National University of Singapore
†{liuchen, lumeiyu, ooibc, shanshan, atung, zhangdo, zmeihui}@comp.nus.edu.sg

ABSTRACT

The online travel services and resources are far from well organized and integrated. Trip planning is still a laborious job requiring interaction with a combination of services such as travel guides, personal travel blogs, map services and public transportation to piece together an itinerary. To facilitate this process, we have designed a cross-service travel engine for trip planners. Our system seamlessly and semantically integrates various types of travel services and resources based on a geographical ontology. We also built a user-friendly visualization tool for travellers to conveniently browse and design personal itineraries on Google Maps.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Design

Keywords

Cross-service, Geographical ontology, Data integration, Ranking

1. INTRODUCTION

Trip planning is often a difficult and time consuming task when a traveller is unfamiliar with the destination. The traveller needs to discover the most attractive sight spots, find the hotels and restaurants, determine the means of transportation and finally confirm the arrival time and departure time at each point of interest(POI). This procedure requires significant search expertise due to the lack of further integration of the online travel services and resources. First, most of the travel sites are domain-closed and only provide specific types of services. For example, Tripadvisor[2] collects user reviews and comments on sight attractions, hotels and restaurants. Expedia[1] provides users with cheap flights and hotels. They do not capture the spatial relationship between different types of POIs, which is important in a trip planning system. Travellers usually select the hotels and restaurants near the sight attractions to save transportation cost. Second, there are tons of reviews and comments associated with the same POI but spreading across different web sites. We observe that many valuable items are either buried or not reachable in the limited search time. Thus, the cleaning and integration of these resources can significantly facilitate the discovery of attractive POIs.

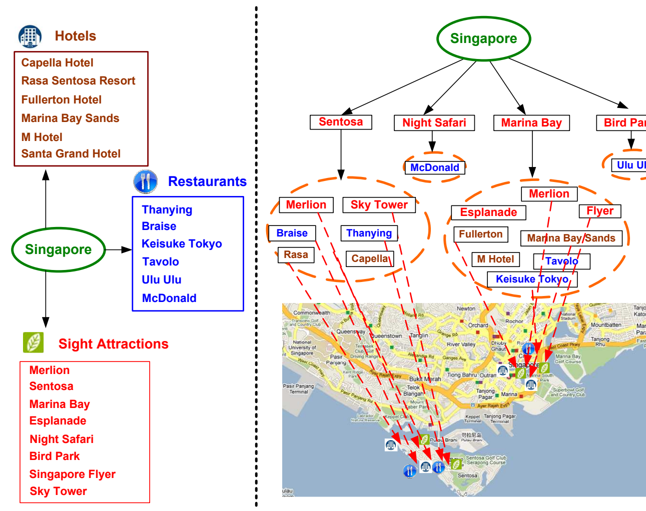

Figure 1: Example of the geographical ontology in Singapore

In order to bridge these gaps in the trip planning, we have designed and implemented a cross-service travel engine which seamlessly and semantically integrates various travel services and resources on the web. The integration relies on a geographical ontology as shown in Figure 1. The left part contains examples of POI in Singapore, including sight attractions, hotels and restaurants. The right part shows the tree structure ontology of these POIs. The child POI is a sub-sight of its parent node. Such a tree structure organization provides better precision and recall in location-based information retrieval[4]. In this example, the ambiguity between "Merlion" in "Sentosa" and in "Marina Bay" can be eliminated. The other advantage is that different types of POIs close to each other are aggregated under the same par-

SIGMOD'11, June 12–16, 2011, Athens, Greece.

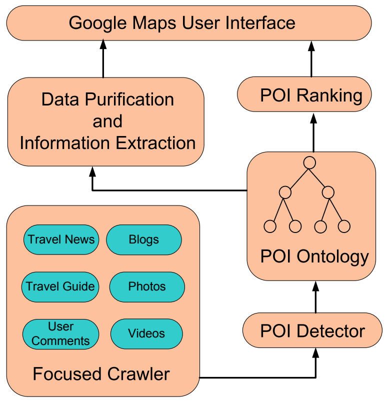

Figure 2: System framework

ent node. The spatial proximity is preserved, which is useful to explore nearby interesting places.

Given this tree-structure geographic ontology, we built a focused crawler to crawl the online travel resources, extract the POIs, and map them to the related POI nodes in the tree. These resources include travel news, travel guides, user comments, personal blogs, photos and videos. Consequently, each POI node in the ontology was associated with a collection of related web resources. These resources were further cleaned and integrated for travel knowledge mining and POI ranking. We ranked the POI in terms of their popularity on the web as well as user comments.

Finally, we implemented a convenient and user-friendly interface for trip planners. They can browse the most attractive POIs and make a trip schedule in a drag-and-drop manner. The whole trip planning procedure is designed as simple as possible so that users can finish a trip schedule in minutes. In summary, we built a cross-service travel engine to facilitate trip planning. It differs from our previous system Marcopolo [4] in that our system in this paper is focused on simplifying the procedure of trip planning while Marcopolo proposed a general framework to support travel resource integrating, searching and sharing in a community.

2. SYSTEM OVERVIEW

The framework of our system is illustrated in Figure 2. We built a focused crawler[3] on the travel topic to fetch the online travel resources. Then, we utilized the POI detector to map the resources to the nodes in the tree-structure geographical ontology. For each POI, we cleansed and integrated the associated information and ranked the POI based on its popularity and user comments. Finally, a map-based interface was implemented to browse and customize the personal itineraries. In the following, we briefly review the design issues of each component.

2.1 Focused Crawler

The focused crawler[3] attempts to download a subset of the world wide web pages which are relevant to a pre-defined topic. The main challenge lies in checking whether a web page is relevant to the specified topic. In our system, we measure the relevance by examining both the global hyperlink graph structure and the parsed html content. A travel dictionary is built by learning from the positive training data sets. Intuitively, if most of the neighboring pages are within travel domain and the page contains a certain number of spatial terms appearing in the dictionary, the web page is considered to be relevant to the travel topic.

2.2 POI Ontology

The geographical POI ontology is required to be complete and unique. In other words, given a POI, there exists one and only one matching node in the tree. We crawled different POI ontologies from several travel sites and merged them to eliminate the duplicate nodes. The POI node with longer path is preferred as it provides more geographical information. For example, "Singapore→ Sentosa → Underwater World" is used while "Singapore→ Underwater World" is discarded during the merge process. Besides mining the tree structure, we also maintained a collection of geo-tags for each POI node as the spatial context used in the POI detection.

2.3 POI Detector

POI recognition from the web resources is a key challenge in many location based applications. Besides examining the textual content and the geographical distribution of hyperlinks[5], we took advantage of the domain specific constraints to improve the precision and recall. The spatial proximity of the detected POIs is important to eliminate the outliers[6]. In addition, the recall can be improved by investigating the spatial context using the geo-tags of each POI. For example, we can infer "Underwater world" from the tag "Dolphin show" in Singapore even when the POI name does not appear.

2.4 Data Cleansing and Information Extraction

As mentioned, each POI was associated with a large collection of related web pages. Users can significantly benefit from the cleansing and information extraction to mine useful travel knowledge such as the admission fee and opening hours of sight attractions, the price and services of hotels and the recommended dishes of restaurants. These information are essentially helpful in the trip planning under limited time and budget constraints. To achieve this goal, we built a SVM learning model on a large training data set which contains user-collected positive examples.

2.5 POI Ranking

The POI ranking based on the online resources is a new and interesting problem. First, it is purely user comment driven without interference from the hidden paid advertisements. Second, the ranking strategies from the service providers could be biased and need to be normalized. For example, the top two attractions in Singapore by TripAdivsor are the "Singapore zoo" and "Botanic Garden" instead of the well known "Sentosa" and "Night Safari". In our system, we ranked the POI in terms of its popularity across the web as well as the related user comments. A POI with high reputation is likely to be visited and reviewed by many users and appear frequently on the web. These user reviews were normalized and aggregated from different sites to eliminate the bias and reflect the true quality of the POI.

3. SYSTEM DEMONSTRATION

In this section, we demonstrate our map interface for users to design personal itineraries. We split the procedure of trip planning into three steps. 1) Select the sight attractions, hotels and restaurants. 2) Make a trip schedule in a drag-and-drop manner to determine the visiting order of POI 3) Determine the transportation approach between two adjacent POIs in the schedule.

Figure 3: Select sight attractions

Figure 4: Select restaurants

Since our demo is targeted at the Chinese travel market, we use trip planning in Bejing as a demonstration example in which the contents are in Chinese. The famous sight attractions in Beijing are displayed on the map, as illustrated in Figure 3. When a sight marker is clicked, the extracted information about the POI are popped out, including the admission fee, opening hours, ranking score based on user reviews, the introduction and the representative photos. These information provide a general idea about the sight attraction and assist users in making the decision. If a user is interested in a sight POI, he can click the "Add To My Sights" button to select the sight as a candidate. Later, our system can recommend hotels and restaurants by considering both ranking score and spatial relationship to the selected sight attractions. Similar to Figure 3, the restaurants and hotels can be displayed on the map and associated with the summary information that users are interested in. For example, we extract and display the average cost, opening hours, recommended dishes and their photos for each restaurant in Figure 4. Users can put their favorite hotels and restaurants into the candidate list by clicking the "Add To My Hotels" or "Add To My Restaurants" button.

Figure 5: Drag-and-drop interface for trip planning

After the selection step, we can get lists of POI candidates of different types. With these candidates, we design a **drag-and-drop** interface to make a trip schedule in a convenient manner. As shown in Figure 5, a planning panel is floating over the map. Candidate sights, hotels and restaurants are listed in the panel. Also, there is a column named "My Trip" in which users can add a new day or delete an existing plan. Users can freely drag the POI from the candidate lists and drop to any day in the "My trip" column. After this step, we can achieve a draft trip schedule. The last step is to find the transportation information based on the POI sequence. We adopt the third-party API service such as Google Maps Direction to search the transportation information between two sequential POIs. Finally, users can flexibly determine the starting date and put the journey into their calendar. Therefore, a trip planning procedure can be finished very quickly and conveniently in our system.

Acknowledgment

The research and system development reported in this paper was supported by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research Grant NRF2008IDM-IDM004-047.

4. REFERENCES

[1] http://www.expedia.com/.

[2] http://www.tripadvisor.com/.

[3] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Comput. Netw.*, 31(11-16):1623–1640, 1999.

[4] Y. Chen, S. Chen, Y. Gu, M. Hui, F. Li, C. Liu, L. Liu, B. C. Ooi, X. Yang, D. Zhang, and Y. Zhou. Marcopolo: a community system for sharing and integrating travel information on maps. In *EDBT*, pages 1148–1151, New York, NY, USA, 2009. ACM.

[5] J. Ding, L. Gravano, and N. Shivakumar. Computing geographical scopes of web resources. In *VLDB*, pages 545–556, San Francisco, CA, USA, 2000.

[6] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *ICDE'10*, pages 201–212, Long Beach, CA, March 2010.