GuideMe! The World of Sights in Your Pocket

Sergej Zerr, Kerstin Bischoff, Sergey Chernov
L3S Research Center, Leibniz Universität Hannover
Appelstrasse 9a, 30167 Hannover, Germany
{zerr, bischoff, chernov}@l3s.de

Abstract—Web 2.0 applications are a rich source of multimedia resources and accompanying metadata, that describe sights, events, whether conditions, traffic situations and other relevant objects along users route. Compared to static sight descriptions, which can be integrated into navigation systems, Web 2.0 resources can provide up-to-date visual information, which has been found important or interesting by other users. Some algorithms have been suggested recently for the landmark finding problem from photos. Still, if users want related videos or background information about a particular place of interest it is necessary to contact different social platforms or general search engines. We demonstrate a mobile application GuideMe! that automatically identifies landmark tags from Flickr groups and gathers sightseeing resources from various Web 2.0 social platforms.

I. INTRODUCTION

When planning a trip people often use various social services to search for photos and videos of representative landmarks near their travel destination. This is possible thanks to the user generated tags, which are especially valuable for recommending, searching and browsing non-textual multimedia resources. However, users usually do not know in advance the main sights in the city of interest and cannot search for it directly. In such situations a system should decide which tags and corresponding photos to return for a given city name. The task of automatic selection of landmarks representative we call the **landmark finding** problem.

Recent research on landmark finding was concentrated on creating landmark photo summaries [1], [2], [3], [4], [5]. Some working prototypes for landmark finding are available online, for example, in the World Explorer application where a user can enter a location name and browse through the landmark related tags and the corresponding photos. However, the photos alone do not provide a complete overview. A tourist might need some background information regarding the landmarks, video guides, recommendations from fellow travellers, etc. Since this information is spread across different social platforms, one has to search for landmarks on each service separately. For example, a user is thinking about going to Hanover and searching for resources tagged with “Hanover”. She wants to see a concise and representative view of the city with a few photos and videos related to famous landmarks. She might also need a couple of web pages with a historical overview and travel tips. Such information need could be satisfied with results from different Web 2.0 services, with photos retrieved from Flickr, videos from Youtube and bookmarks from Del.icio.us, like in Figure 1.

![Fig. 1. Landmark resources for “Hanover”.](http://de.wikipedia.org/wiki/Marktkirche_(Hannover)

This scenario requires several searches on different social platforms, which might be particularly inconvenient if a user is already travelling and looking for the interesting sights using her mobile phone. It is also challenging to filter landmark-related information on each service, as this problem is currently addressed only by photo sharing providers.

In this work we demonstrate a mobile application GuideMe! that retrieves landmark-related resources from various Web 2.0 platforms. First, GuideMe! extracts representative tags from landmark pictures according to the algorithm which we earlier described in [5]. The landmark-classified tags are used to query a set of social sources like Flickr, Youtube, Delicious, Slideshare, etc. Finally, the retrieved results are fused and ranked according to their relevance, popularity, rating and number of comments. Contributions of this demonstration include: (i) an algorithm for efficient extraction of landmarks from Web 2.0 sources for a given location; (ii) a system for federated search of Web 2.0 resources related to these

1http://tagmaps.research.yahoo.com/worldexplorer.php
2www.flickr.com
3www.youtube.com
4www.delicious.com
landmarks; (iii) a ranking strategy to provide users with a representative overview of the up-to-date sightseeing information.

We present the system architecture in Section II, describe our method for landmark finding/tag extraction and the dataset used in Section III. In Section IV we present an overview of the demonstration and then conclude by summarizing our main contribution and addressing open questions.

II. SYSTEM ARCHITECTURE

This section describes the three main components of the GuideMe! system architecture, see Figure 2. In the following we will present the GUI of the mobile client, followed by the landmark extractor web service and finally the InterWeb web service - a mashup network which integrates a number of tools like Flickr, YouTube, Slideshare, etc.

A. Mobile Client

The GuideMe! application can be installed on any mobile device with Android \(^5\) operating system. The graphical user interface consists of the explorer page and the settings page, see Figure 3. The explorer page contains a search field and a list of results is displayed after successful search query execution. At the settings page the user can define how many resources should be displayed in the result list and select their preferred data types. The available data types are:

- Videos (from YouTube, Vimeo and Ipernity)
- Pictures (from Flickr and Ipernity)
- Presentations (from Slideshare)
- Bookmarks (Delicious)

Each result can be previewed at the original web page using the integrated browser. GuideMe! uses an internet connection over WLAN or GSM, dependent on their availability, in order to interact with web services that wrap other application components.

\(^5\)http://en.wikipedia.org/wiki/Android_(operating_system)

B. Landmark Seeker Webservice

The Landmark Seeker is a SOAP\(^6\) based web service that provides an interface with the method getTopKcityTags with the mandatory (String) parameter cityName.

The response is an XML-formatted list of sightseeing labels corresponding to the given cityName. For example the query hanover could return the following list: “herrenhausen”, “nordlb”, “rathaus”, “cityhall”, and “marktkirche”. The web service implementation is Java based (JDK 6) by using the Apache CXF\(^7\) Framework. Its WSDL is available online\(^8\).

C. InterWeb

InterWeb is a web service which integrates a number of different Web 2.0 tools like YouTube, Flickr, Ipernity, Slideshare, and Delicious. Most of the Web 2.0 applications and their orchestrations focus on finding resources related to user information needs. Portals like iGoogle and Netvibes also help to locate information distributed across different information sources. However, such portals typically provide no facilities for integration/merging of information obtained from these sources without providing a way of actually linking them together.

InterWeb provides a rich set of functions and a seamless overview of the entire set of distributed Web 2.0 resources. In this manner InterWeb serves as a Meta-Web 2.0 service. It provides a uniform interface for basic functionalities such as search. InterWeb is a PHP based implementation and is available online\(^9\).

\(^6\)http://www.w3.org/TR/soap/
\(^7\)http://cxf.apache.org/
\(^8\)http://pharos.l3s.uni-hannover.de:9966/landmarkseeker?wsdl
\(^9\)http://athena.l3s.uni-hannover.de:8000
III. LANDMARK DATA EXTRACTION

To provide users with a set of landmark resources we first need to identify the tags associated with landmarks. Our method for extracting landmark information from Web 2.0 exploits tags and social groups from Flickr. The full details of the landmark tags extraction can be found in [5] and here we provide a quick summary of our approach. One important difference to the original paper is the change of classifier. While in [5] we used the SVM-Light package, here we chose the NaiveBayesMultinomial classifier from WEKA10 as it builds models faster and delivers comparable classification results.

A. Step 1: Classification of Landmark Photos

In the beginning we select a set of photos related to a particular city. Here we rely on a simple heuristic of having the city name as a tag associated with a photo. From the set of pictures containing a city tag, we want to select photos representing landmarks. For training the classifier we pick photos from existing landmark-related Flickr groups. The idea is that some of the Flickr groups like “Landmarks around the world” can serve as positive examples, while arbitrary general groups, like “Birds” or “Airplanes” represent negative examples. For the current demonstration we used about 50,000 photos from Flickr groups about landmarks and 100,000 photos from groups about other topics. A classifier assigns each photo to either the “landmark” or the “non-landmark” category given the input feature vectors. Each tag gets a weight equal to its normalized tag frequency \( TF_r(t) \). In the following definitions \( U \) represents the set of users, \( T \) stands for the set of tags, and \( R \) is the set of resources. Let \( T_r \) denote the set of tags assigned to a resource \( r \). Since in Flickr each tag can appear only once per image the normalized tag frequency \( TF_r(t) \) of a tag \( t \) in a resource \( r \) is defined as follows:

\[
TF_r(t) = \frac{1}{|T_r|}
\]

(1)

In an experimental evaluation we achieved 98.92% accuracy on training data (0.999 AUC/ROC) and 97.37% accuracy (0.996 AUC/ROC) when applying 10 fold cross-validation.

B. Step 2: Ranking of Landmark Tags

Once we have selected a set of city photos and filtered only landmark-related ones, we want to see those tags representing landmarks specific to a particular city. We would like to give low scores to common tags and high scores to city-specific tags. The assumption is that representative landmark tags appear in landmark photos, but are not very common among the whole collection of images. Let \( R_t \) denote the set of resources which contain tag \( t \) and \( U_t \) be a set of users who used tag \( t \). Similar to the notion of Inverse Document Frequency in Information Retrieval we define Inverse Resource Frequency and Inverse User Frequency as follows:

\[
IRF(t) = \log\left(\frac{|R|}{|R_t|}\right)
\]

(2)

\[
IUF(t) = \log\left(\frac{|U|}{|U_t|}\right)
\]

(3)

We compute \( IRF(t) \) (Eq. 2) and \( IUF(t) \) to penalize popular tags. If a tag is frequently used to tag photos in the dataset, it has a low \( IRF(t) \) value and vice versa. Similarly, if a tag is globally very common amongst users, it must be scored low. This is achieved by computing \( IUF(t) \) (Eq. 3).

After defining global scoring factors, we come to local measures computed on that part of the collection with landmark photos only. When considering the dataset containing only pictures associated to a particular city and classified as landmarks, our assumption is that common tags should be scored high. Let us represent the set of landmark-related photos selected for a city as \( R_c \) and the corresponding tag set as \( T_c \). If a tag is common among the photos for a particular city as well as among many users, probably this tag represents some feature of the city, e.g. some museum, or an old and famous building. Let \( U_c(t) \) be a set of users using a tag \( t \) for the landmark photos for a city \( c \). We compute the normalized City Tag Frequency, \( CTF \), using (Eq.4):

\[
CTF(t) = \frac{|U_c(t)|}{\text{MAX}(|U_c(t')|)}
\]

(4)

We combine all the above mentioned factors that affect the ranking of the tags and compute a representativeness score for each tag \( t \) occurring along with the resources classified as landmarks of a city \( c \). Here, the final scoring function as proposed in [5] has been adapted to better fit the new, though, similar dataset and demo scenario, e.g. with respect to time constraints (i.e. the number of pictures queried for from Flickr). The representativeness score of each tag for a city \( c \) is computed as follows:

\[
\text{SCORE}(t) = IRF \cdot IUF \cdot CTF
\]

(5)

Therefore, each city is assigned a ranking of landmark related tags and this information is stored in a database. Both Step 1 and Step 2 are performed offline and do not affect system performance.

C. Step 3: Collecting Landmark Resources

When calling the web service, first a database with previously learned city tags is queried for the given city name. If there are city tags – already learned according to the steps described above – in the database, the top-k city tags will be returned in descending order. If no city tags have been learned for the city yet, a search request with the name will be sent to the Flickr API. This request returns all relevant photos that have been tagged with the given city name. Then the classification and tag ranking procedures presented above are applied on the returned set of photos. The top 20 city tags will be saved to the database and the top 5 of them will be send to the client. Employing this method [5] obtained 12% improvement in precision over the World Explorer system.
D. Step 4: Ranking Landmark Resources

Using the set of landmark tags returned by the seeker service, InterWeb combines them into a query and executes it on available Web 2.0 services. In order to provide a representative and diverse overview of sights in Hanover on a small display, the results have to be ordered before presenting them to the user. The list of sightseeing results returned by InterWeb is first divided into sub-lists, one for each landmark. We sort each of the lists by relevance to the corresponding landmark and the top-ranked result from each list is displayed to a user. If some result is relevant to several landmarks it is copied to each corresponding list, see Fig. 4.

The relevance depends both on the content of the resource and its popularity at the source service. The popularity can be derived from the number of views, comments and ratings of the resource. Thus the relevance can be computed as a weighted sum of these factors. Let us define $\text{Sim}(r, t)$ as the textual similarity between resource $r$’s description and tag $t$, $V_r$ as the number of views of resource $r$, $C_r$ as the number of comments assigned to $r$, $R_r$ as a rating assigned by users, and $P_r$ as the position of $r$ in results ranking returned by a particular social service. We rank all landmark resources according to their relevance to a landmark using the Equation 6:

$$\text{RELEVANCE}(r) = \alpha \cdot \text{Sim}(r, t) + \beta \cdot V_r + \gamma \cdot C_r + \delta \cdot R_r + \epsilon \cdot P_r,$$

(6)

where $\alpha$, $\beta$, $\gamma$, $\delta$ and $\epsilon$ are coefficients used for tuning the system.

IV. DEMONSTRATION OVERVIEW

In this demonstration we will primarily show how GuideMe! works and sightseeing resources of different data types can be retrieved. We will demonstrate the resource discovery process using a mobile device. Additionally we will explain the complete query process including landmark extraction and resource aggregation at InterWeb.

First the user selects the preferred data types such as images, videos, or presentations at the options page and saves these settings. She types the search terms into the text field at the exploration page and starts the execution. The results are shown as a ranked list and the user selects the resource of her interest and can view it using the integrated browser.

The demonstration application package can be downloaded from our page and can be installed on any mobile device with android operating system or an android emulator available for PC.

V. CONCLUSIONS AND FUTURE WORK

Web 2.0 resources and metadata are very valuable for answering the diverse and complex information needs users have. In this demonstration we focus on a landmark finding scenario. We identify and extract landmark information from multiple social platforms and compile a representative summary for a given city. We present a mobile search interface which retrieves landmark resources from sites like Flickr, YouTube, Delicious, etc. and fuses them. Our contributions include an algorithm for efficient extraction of landmarks from Web 2.0 sources, a system for federated search of Web 2.0 resources related to these landmarks and a ranking strategy to provide a user with a representative and diverse overview for sightseeing.

An open question is what is the best type of resource for a particular landmark. Some static objects like buildings or paintings look good on photos, while objects like church bells call for a video representation. In future we would like to explore how well different types of resources are suited to visualize specific types of sights.

ACKNOWLEDGMENT

This work was partially funded by the NTH School for IT Ecosystems. NTH (Niedersächsische Technische Hochschule) is a joint university consisting of Technische Universität Braunschweig, Technische Universität Clausthal, and Leibniz Universität Hannover. The authors would like to thank Nicole Ullmann and Philipp Kemkes for their contributions in implementing this demo application.

REFERENCES


11http://l3s.de/~zerr/guideme/