Discovering Local Attractions from Geo-Tagged Photos

Ombretta Gaggi
Department of Mathematics
University of Padua
via Trieste, 63, 35131 Padova, Italy
ombretta.gaggi@unipd.it

ABSTRACT

This paper presents *PhotoTrip*, an interactive online service which is able to recommend not widely known, but interesting, places along travel itineraries, without human intervention. These points of interest are identified by analyzing the geo-tagged pictures contained in Flickr community. *PhotoTrip* enriches these pictures with information provided by WikiLocations, to enhance the recommended local attractions with additional information on what is depicted in the photos to help the user to make a choice. We evaluated the system with an user study.

Categories and Subject Descriptors

H.3.5 [**On-line Information Services**]: Web-base Services; H.5.1 [**Multimedia Information Systems**]: Hypertext navigation and maps

General Terms

Algorithms, Experimentation

Keywords

photo collections, geolocalization, knowledge discovery

1. INTRODUCTION

A key issue for success of a travel to a new city is to have a good plan for the trip. For this reason, travelers search for information in guidebooks, popular web sites or blogs. All these options have shortcomings: travel books do not cover all cities/locations in the world, are not free and may not have the right level of detail. Personal travel blogs report a particular traveler's view, which may disagree with the experience of other travelers, and can depend on the amount of preparation invested in planning the trip. Moreover, searching blogs for information is a task which is time consuming and may require significant search expertise.

There is another particular situation: the traveler has a clear idea about the source and the destination of his/her trip, but he/she wants to know if, along the path to reach the destination, there are some local attractions, which are not widely known, but still of interest. We call these places LPOIs, *Local Points Of Interest*. If you travel from Padua to Venice in Italy, an example of *LPOI* is Stra, a little city, not so famous as Venice, but there is *Villa Pisani*, a fine villa of the Venetian noble Pisani family, which requires a deviation of less than 5 km from the original route.

This paper presents *PhotoTrip*, an online service which is able to recommend points of interest along a tourist route, without human intervention. These points of interest are identified by analyzing and clustering the geo-tagged pictures contained in Flickr, the leading online service for managing and sharing photos. This work represents a new way to use the many geo-tagged and time-stamped photos available on the Web: to enhance the experience of each traveler discovering local attractions in the real world. Moreover, our system can help to promote places that are not widely known, but still interesting.

Our system improves information provided to the users by other services: Google Maps, for example, shows "interesting" photos which had been taken in the same area of a required route, but it does not allow to calculate deviations and does not filter images with respect to a particular category requested by the user (e.g., nature, architecture, etc.) or the maximum deviation allowed. Google Maps does not recognize point of interest, but simply displays all the interesting pictures taken in the whole rectangular area depicted in the web page.

PhotoTrip searches for LPOIs within a maximum deviation given as input by the user. Then, in order to better help the user to make the correct choice, *PhotoTrip* integrates other online services, like WikiLocations, to enhance the information provided for the recommended local attractions with additional information on what is depicted in the photos. Finally, once the user has chosen the LPOIs, it calculates the route showing the directions to follows.

The paper is organized as follows: Section 2 discusses similar approaches in literature. Section 3 describes the system and its implementation. A case study is discussed in Section 4. We conclude in Section 5.

2. RELATED WORK

Other works in literature address the problem of mining travel route from geo-referenced photos. Crandall et al. were the first to map images collected from Flickr to infer a relational structure [3]. They automatically identify most photographed places, therefore a set of Points of Inter-

SAC'13 March 18-22, 2013, Coimbra, Portugal.

est (POIs). Moreover they suggest to create an online travel guidebook that could automatically identify the best sites to visit during vacation, as judged by the collective wisdom of the world's photographers.

The problem of mining travel itineraries from photos collections taken from web services like Flickr or Panoramio has been investigate in [4] and [10]. De Choudhury et al. [4] analyzed geo-tagged collections of pictures taken from a single users to estimate the travel path he/she followed. Extracted itineraries followed by a big number of users are then merged into a POIs graph which is used to construct suggested travel paths for several major cities. An extensive user study has shown the high quality of the automatically generated itineraries.

Popescu and Grefenstette [10] implemented a recommendation system for tourist visits based on the aggregation of photos' annotations. They mined the record of visited landmarks to build a user-similarity matrix. Then they produce a lists of possible interesting attractions based on the experience of like-minded users.

Geo-tagged photos available in social media are mined also by Arase et al. [2], Kurashima et al. [8] and Lu et al. [9]. Arase et al. [2] defined a set of trip themes, i. e., visiting landscape or cities, historical building or modern art, and mined frequent trip patterns for each theme category. Kurashima et al. [8] presented a framework for travel route recommendation which is able to suggest different travel path according to the spare time of the user. The system is based on the analysis of both geo-tagged and time-stamped photos of photo sharing services. Even in this case, the authors mined the users' pictures to learn personal travel histories using a probabilistic approach.

Photo2Trip [9] is an automatic travel route planning which collect photos from Panoramio in order to suggest customized trip plans according to users' preferences. It is able to suggest the most popular destinations to visit, and a time arrangement. Differently from other approaches, the system allows the users to interactively specify their preferences.

All the cited systems address the problem of identifying (and defining) the most popular POIs and to create a travel path with them. Our approach is different since the user already knows where he/she wants to go and what he/she wants to visit, but he/she wants to investigate if there exist some minor, but noteworthy, local attractions within a predefine deviation limit. Moreover, our system retrieves additional information from *Wikilocation* to help the user's choice.

Kofler et al. address a similar problem in [7]. *Near2me* is a prototype system that allows to recommend places that are not necessary touristic but are genuinely representative of the considered places. Therefore *Near2me* allows to extract POIs under a different perspective.

3. SYSTEM DESCRIPTION

We may summarize how *PhotoTrip* works, at high level, in the following steps:

1. the user fills the form data to start the search. The system asks the user the start and the end address[1], the maximum deviation allowed, the means of travel (by car or by feet) and what the user is interested in[2];

2. the client checks data consistency. If the data are correct, they are sent to the server, which calculates the path and the search area. To minimize waiting time, the system implements a *"divide et impera"* policy: the search area for LPOIs is divided into a grid of boxes, where each box is identified by two points a and b, representing, respectively, the upper-left and the right-bottom angles of the box. Each box is associated to a sub problem of the original problem, i. e. a search into a smaller area. Then, for each sub problem, the communication between client and server proceeds in the following way:

   (a) the client sends the request to Flickr for the proper box;

   (b) the server retrieves the results and returns them to the client;

   (c) the client fills a photo gallery with the received results. Each sub problem contributes to the final result, depicted in Figure 2, if a LPOI is found. The user can play a presentation of the pictures regarding LPOIs found so far, or can choose a single photo;

   (d) the route is complete when all the sub problem ended their computation.

3. At the end of the computation, the system notifies the user (see Figure 2).

All the sub problems are completely independent, therefore can be executed in parallel.

We divided the search in sub problems to give a (partial) output to the user as soon as possible. Step 2c describes this choice: the user can interact with pictures and LPOIs as soon as the computation for the first search box ends up or the first LPOIs is found. Then, the server adds to the gallery other photos of LPOIs found in other boxes, till the end of the computation.

3.1 Definition of the problem

To better explain the algorithms which implement *PhotoTrip*, we need to describe the data model used in this work. Our system retrieves collections of pictures from Flickr and

[1]The radial search requires only the start point and searches within a maximum radius.

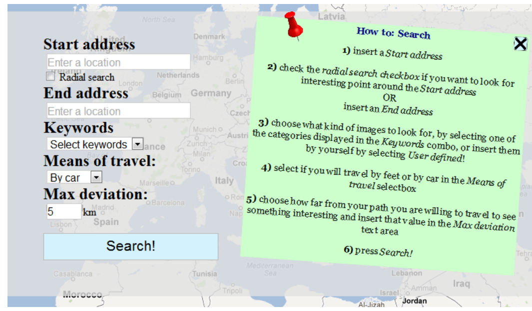


Figure 1: Screenshot of *PhotoTrip*: the initial form.

[2]The system contains some predefined categories, e.g., architecture (containing the keywords palace, building, villa, caste, etc), nature (containing nature, tree, mountain, etc), recent photos (which simply select the newest photos), etc.

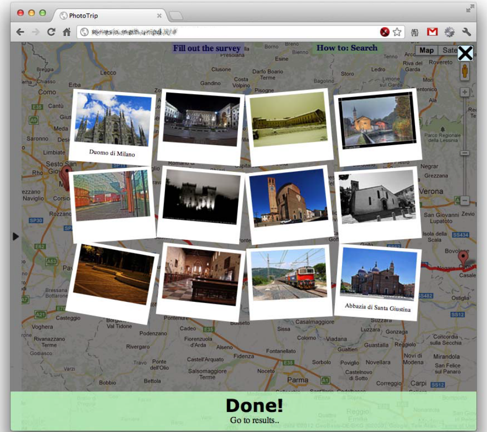


Figure 2: Screenshot of *PhotoTrip*: gallery.

filters them on the basis on a set of criteria. A collection of photos is a set $C = \{p_i\}$ where each p_i is a tuple

$$\langle id_{p_i}, geo_{p_i}, T_{p_i}, fav_{p_i}, v_{p_i}, c_{p_i} \rangle$$

where:

- id_{p_i} is a unique identifier;
- $geo_{p_i} = \langle lt_{p_i}, lg_{p_i} \rangle$ is the location in which the picture was taken. It is represented by a tuple containing the latitude (lt_{p_i}) and the longitude (lg_{p_i}) of the photo's location;
- $T_{p_i} = \{tag_j\}$ is the set of tags of photo p_i;
- $fav_{p_i} \in \mathbb{N}$ is the number of users who add p_i to the set of favorite photos;
- $v_{p_i} \in \mathbb{N}$ is the number of visualizations for p_i and
- $c_{p_i} \in \mathbb{N}$ is the number of comments posted for p_i.

To retrieve photos from Flickr, the system compares each T_{p_i} with a set of tags T_u defined by the user through the web interface. The system allows the user to choose between a set of predefined categories (e.g., architecture, nature, etc) or to input a set of tags. For each category, a built-in set of tags is defined, calculated a priori on the basis of further analysis of the Flickr data set.

Photos are retrieved on the basis of the set T_{p_i} and of the geolocalization geo_{p_i}. Then the obtained collection of photos C, which represents photos within the maximum deviation defined by the user and with a correct set of tags, is analyzed and filtered to retrieved a list of LPOIs for each box in the route. To this end, the first step is to filter the collection C to select only relevant pictures, then the remaining set of pictures are clustered into LPOIs.

The first step was initially computed by analyzing how many times an user marks a particular photo as favorite (fav_{p_i}), how many time a photo was visualized (v_{p_i}) and the number of comments it received (c_{p_i}). A combination of these three number must be greater than a threshold, statistically calculated analyzing photos data set in Flickr community.

This first approach has highlighted the following limits. To obtain the information regarding the number of users who have added a particular photo to his/her favorites the system needs to make a further request to Flickr API, i.e., it requires more time. In the same way, to obtain the number of comments of a photograph requires extra requests and extra time. We must note here that, unlike other recommended system for touristic travel, our goal was to develop an on-line system and Flickr API allows a maximum of one request per second. Therefore, the implementation of a filter based on fav_{p_i} and c_{p_i} requires, at least, two additional seconds for each analyzed picture, i. e. an excessive increase in processing time.

Moreover, a deeper analysis shows that the number of "favorites" and comments of a picture is not a good representative of the degree of interest of a picture, because it is often related to the particular user: e. g. an user can upload a picture of his/her children and mark it as favorite, but this information is useless, if not harmful, for our system.

It was therefore decided to remove the analysis of the number of comments received and the number of times that picture has been marked as favorites, and to develop a filter based on the number of visits received from each photo. The threshold is calculated dynamically for each set of pictures to be sure to always return a list of LPOIs. In this way, the time need for each search was lowered while maintaining a good quality and quantity of the photos displayed by *PhotoTrip*.

At this point, the selection of pictures is complete. As already discussed in the beginning of Section 3, the photo selection process is performed for each box in which the path is divided. Each box is identified by two points a and b, representing, respectively, the upper-left and the right-bottom angles of the box. Given the set $box_{a,b}$, which represents the infinite set of points contained in box identified by a and b and a set of tags T_u given as input by the user, we can define the obtained collection of pictures as

$$\begin{aligned} C^f_{a,b} = \{p_i| \quad & geo_{p_i} \in box_{a,b} & \wedge \\ & \exists j | tag_j \in T_{p_i} \cap T_u & \wedge \\ & v_{p_i} > threshold_{a,b} & \} \end{aligned}$$

where $threshold_{a,b}$ is calculated as the arithmetic mean of the visits received by each photos in $box_{a,b}$.

For each box, if the returned collection $C^f_{a,b}$ is non empty, the second step is to aggregate this pictures into LPOIs.

To this end, the system aggregates pictures according to their position on the map. All the pictures with a distance less than a threshold are considered as belonging to the same LPOI. An LPOI contained in $box_{a,b}$ can be defined as a tuple

$$LPOI = (lt_{p_c}, lg_{p_c})$$

where lt_{p_c} and lg_{p_c} are, respectively, the latitude and the longitude of a picture chosen as geotag for that particular LPOI. Given a candidate photo p_c for an LPOI and a threshold distance δ, the set of pictures contained is defined as:

$$\{p_i | p_i \in C^f_{a,b} \wedge |p_c - p_i| \leqslant \delta \ \}$$

where δ is calculated as the minimum value between the maximum deviation defined by the user and 400 meters and the candidate photo p_c is the picture with is a photo depicting the LPOI with the lowest value of longitude.

More in detail, the set of images is aggregated into LPOIs in the following way:

1. the set of photos is sorted according to their longitude;

2. the first picture is removed from the set and is considered a candidate picture p_c for a new LPOI;

3. all the photos with distance less then δ from p_c are removed from the set and added to the new LPOI;

4. step 2 and 3 are repeated until the initial set of photos is empty.

We must note here that the candidate photo for an LPOI p_c is not considered a representative photo for that particular LPOI from a semantic point of view and it has not a particular relevance in the visualization of the LPOIs, where all the pictures are displayed in a balloon (see Figure 4). It is simply a starting point for the computation.

The set of LPOIs returned to the user is the union of all LPOIs found so far.

3.2 System Implementation

To improve efficiency and scalability of the system, we designed *PhotoTrip* moving the data processing, where possible, on the client side, and the server is used only to retrieve photos from Flickr, to manage the cached routes and additional information (i. e., link to Wikipedia). The system is obtained through the integration of different multimedia services available on the web, i. e. Google Maps, Flickr and WikiLocation [5]. Each time an user searches for a travel route, the system:

1. retrieves the shortest path from the starting address to the destination from Google Maps;

2. calculates the division of the path in boxes;

3. for each box retrieves and filters photos from Flickr community and fills the photo gallery;

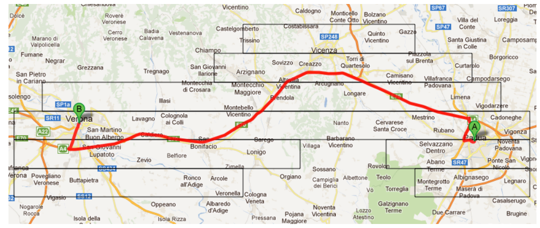
4. retrieves information on what is displayed in the photos from WikiLocation and

5. calculates and shows the suggested path.

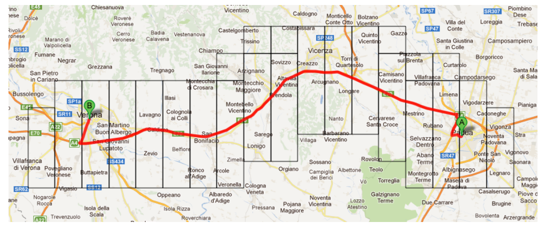
The segmentation of the route from the starting address to the destination is calculated using a customized version of RouteBoxer [6], a JavaScript library developed by Google with the aim to manage travel paths.

The area covered by the route is initially identified by a rectangle that includes the departure and destination points. This rectangle is divided into a grid where each squared cell has the side equal to the maximum deviation accepted by the user. A second step marks the cells which contain a route segment and the adjacent ones as the area to search for LPOIs.

The system combines the cells, which are adjacent in the area returned by the previous step, to create larger boxes. This step is particularly important, since a big number of boxes means a big numbers of calls to the Flickr API, i. e., a large response time of the system. On the other hand, a small number of boxes means that the average size of the boxes increases and this may affect the efficacy of the tool to identify LPOIs. As an example, boxes which contain famous cities like Venice or New York do not allow to identify minor attractions in the same area because there is a big number of frequently visited photos related to those cities and very few photos related to the rest of the area. This problem is solved if the boxes are sufficiently small to include only the city or a part of it[3]. Therefore, the tool must find a good tradeoff between the efficacy brought by a big number of small boxes and the response time of the overall system.



(a)



(b)

Figure 3: Travel path segmetation by: (a) RouteBoxer, (b) *PhotoTrip*.

The adopted solution is to merge each square, with side equal to the maximum allowed deviation, with its adjacent ones, once in a horizontal direction, and once in the vertical direction. We obtain two sets of rectangles, where each rectangle has one side equal to the maximum allowed deviation. Then, the set with smaller cardinality is chosen for the analysis of the contained photos.

Figure 3 shows the sets of boxes calculated by RouteBoxer (Fig. 3(a)) and by our algorithm (Fig. 3(b)). We must note here that RouteBoxer creates larger boxes which reduce efficacy of the system. Many tests have shown that our solution returns segmentation into rectangles more effective to find LPOIs both for horizontal, vertical and diagonal paths.

The last step of the travel path segmentation sorts the boxes to search for photos first in the even boxes and then in odd ones to populate the initial gallery faster.

Photos collected for each LPOI are displayed using popup balloons (see Figure 4). The user can browse the pictures or play a built-in multimedia presentation with a slideshow with transitions between different photos. The user can select a LPOI as a waypoint: in this case the system calculates the new travel path. Moreover the system calculates also a "proposed travel route" which contains all the identified LPOIs. The user can also mark a picture as "not relevant". This information is stored in a database and the system does not select that image anymore as soon as the number of these reports exceed a threshold.

Pictures are not always sufficient to choose a waypoint. For this reason, *PhotoTrip* retrieves text information about photo location from WikiLocation, that allows searching on Wikipedia articles that have been geo-located within a certain distance from a geographical point (see in Figure 5). We have evaluated also GeoNames [1], a similar web service which returns a larger set of articles as result, but allows only

[3]This discussion can be scaled also to cities' districts if the search is performed inside a single city.

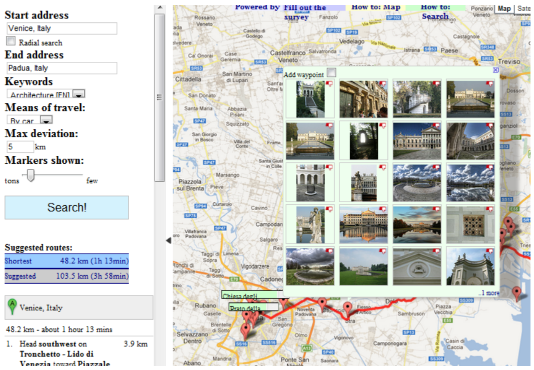


Figure 4: Screenshot of *PhotoTrip*: Villa Pisani, a local attraction.

Figure 5: A detail of the initial gallery with information retrieved from Wikilocation.

a limited request number a day. Therefore the system first searches WikiLocation for appropriate articles, if no result is found, the query is performed also on GeoNames.

The system was able to automatically associate text information to 2,500 photos over 48,000, i. e., about 5%. This is not a poor result if we consider that cameras usually register the position of the photographer and not of what is depicted in the taken photos, which can be far away. Moreover, many photo does not have a title, a description, or a set of detailed tags. Returned articles are stored in a cache.

PhotoTrip allows also a radial search which returns local attractions within a maximum deviation from a starting point defined by the user.

4. USER EVALUATION

The system has been available online to users since May 2012. The following data were collected during a period of 30 days: 110 users have used the service, they were located mainly in Italy, but there have been some access also from the United States, France and from England. The average time spent by each user within *PhotoTrip* is about 6 minutes. The system calculated 880 different travel paths, i. e., on average, each user requests 8 different routes. All these data show a high interest from users since the average time of permanence and the high number of requested travels path per user shows that users have interacted with the site for a quite long time.

Data collected shown that the users accessed to the system with different browsers: 32% used Google Chrome, 10% Firefox, 2% Safari, and 2% Internet Explorer. No problem has been reported with any browser.

We ask the users to answer to a questionnaire of nine multiple choice questions about the experience made with

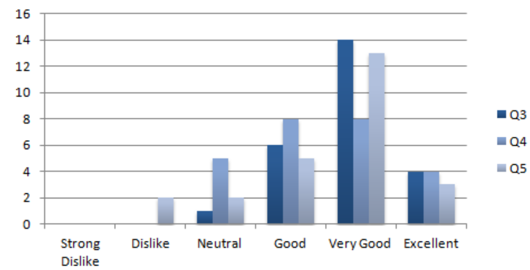


Figure 6: Answer to questions Q3:"How do you rate the usability of the system?", Q4:"What do you think of the *PhotoTrip*'s layout? Colors, shapes, animations used to display photos, etc.." and Q5:"How to you rate the system ability to associate a text description to the retrieved photos?".

PhotoTrip. The users rate the answers on a scale from 1 to 6 (1 = "strong dislike", 2 = "dislike", 3 = "neutral", 4 = "good", 5 ="very good", 6="excellent"). A last open question gave the possibility to express additional remarks. 25 users answered to our questions.

The questionnaires revealed the following insights. The result was positive, since only one user has reported some difficulties with the *PhotoTrip*'s interface, the remaining 24 users instead, gave a positive assessment of the usability of the system, in particular 18 out of 25 users, i. e., 72% consider it very or extremely intuitive[4] (see series "Q3" in Figure 6). Even the graphics are rated positively by all users, only 5 users rated the overall appearance as "neutral", the remaining 20 over 25, i. e., 80% rated the graphics more then adequate (8 users rated the appearance as "good", 8 users as "very good" and 4 as "excellent") as depicted by series "Q4" in Figure 6.

We asked to the users to evaluate the capacity of the system to find out a description about the position of the attractions (and the attractions themselves) depicted in the returned set of photos for each LPOI through WikiLocations and GeoNames. The answers reported that only 8% of users were unsatisfied with this feature, the remaining 92%, i. e., 23 out of 25 users, gave a positive remark; among these 16 users rated it as "good" or "very good" as reported by series "Q5" in Figure 6. Moreover, we asked to the users to evaluate the relevance of pictures returned from *PhotoTrip*: no user reported a negative feedback, 92% of the users judged positively the number of returned pictures and LPOIs and appreciated the relevance of the results compared to the selected keywords: only 2 users expressed a "neutral" mark as depicted by series "Q8" in Figure 8.

Finally the user rated very positively the overall system, 100% of the feedbacks were positive and 72% or the users rated *PhotoTrip* as "very good" or "excellent" (see series "Q9" in Figure 8).

5. CONCLUSIONS

In this paper we have presented *PhotoTrip*, an interactive online service to discover not widely known, but interesting, places along travel itineraries. These local points of interest

[4]I. e., they rated the usability of the system as "very good" or "excellent".

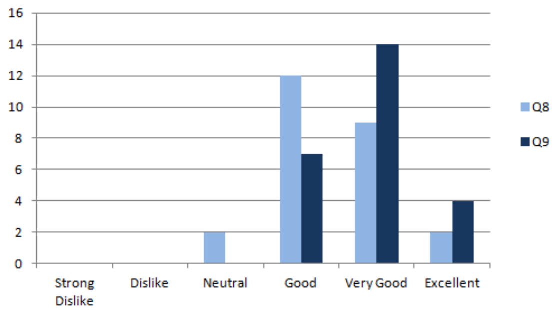


Figure 7: Answer to questions Q8:"How do you rate the relevance of the returned photos and LPOI with the initial search?" and Q9:"How do you rate *PhotoTrip*?".

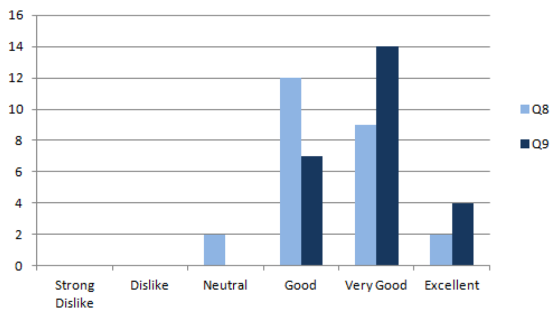


Figure 8: Answer to questions Q8:"How do you rate the relevance of the returned photos and LPOIs with the initial search?" and Q9:"How do you rate *PhotoTrip*?".

are identified through the analysis of the photographs contained in the Flickr community. Since *PhotoTrip* generates travel paths in a fully automatic way, it can be used to discover LPOIs anywhere in the world thank to amount and distribution of data available, about 350 million geo-tagged photos, in Flickr. Moreover, it also allows users, who are interested in, to obtain additional information on what is portrayed in the photos themselves, thus meeting the thirst for knowledge of the most curious visitors.

A user study has shown that the user deeply appreciated the system and the set of LPOIs which it is able to discover. Moreover, the users rated very well the system's usability. The system is accessible through different devices, i. e., desktop, portable pc and tablet. We plan to improve the interface for smartphones since, due to the small size of the screen, some interaction widgets become too small.

Although the use of Flickr community allows searching everywhere, we must note that the results returned by *PhotoTrip* are strongly influenced by the language used to define the tags, both tags which the photographers associate to the pictures taken, and the tags defined to start a query by the user of the system (or by the system itself, in the case of predefined categories).

Photographers usually describe their photos using their mother tongue and, at a later time, they sometimes add some translations, most frequently in English. This means that the language used in the tags which describe a photos depends on two factors, the mother language of the photographer and where the picture is taken, but the first factor is more important of the second one: e. g., there is not guarantee that pictures taken in Italy contain Italian tags. For this reason, at the moment, we implement two sets of predefined categories, one using English language, and a localized set in Italian.

Our future works will investigate the possibility to increase the number of retrieved photos integrating an online translator service to translate the tag in English, in the case of tag defined by the user, in the language used in the country of the searched travel path and, if available, in the native language of the photographer who uploaded the photo. This issue is particularly important since the same problem about the language affects also the efficacy of the system to find information about the LPOIs using WikiLocation or GeoNames, which contain a bigger number of English articles with respect to other languages.

Acknowledgment

The author would like to thank Michele Torresin for the development of the system. Partial financial support for this work is provided by MIUR/PRIN ALTER-NET and the UNIPD/PRAT Web Squared projects.

6. REFERENCES

[1] Geonames. http://www.geonames.org, 2012.

[2] Y. Arase, X. Xie, T. Hara, and S. Nishio. Mining people's trips from large scale geo-tagged photos. In *Proceedings of ACM MM 2010*, pages 133–142, 2010.

[3] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Proceedings of WWW '09*, pages 761–770, 2009.

[4] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *ACM conference on Hypertext and hypermedia*, pages 35–44, 2010.

[5] B. Dodson. Wikilocation, the geolocation api for wikipedia. http://wikilocation.org/, 2012.

[6] Google, Inc. Routeboxer. http://google-maps-utility-library-v3.googlecode.com/svn/trunk/routeboxer/docs/examples.html, 2012.

[7] C. Kofler, L. Caballero, M. Menendez, V. Occhialini, and M. Larson. Near2me: an authentic and personalized social media-based recommender for travel destinations. In *Workshop on Social media*, pages 47–52, 2011.

[8] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *Proceedings of ACM CIKM '10*, pages 579–588, 2010.

[9] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *Proceedings of MM '10*, pages 143–152, 2010.

[10] A. Popescu and G. Grefenstette. Mining social media to create personalized recommendations for tourist visits. In *Conf. on Computing for Geospatial Research & Applications*, pages 37:1–37:6, 2011.