How Random Walks can Help Tourism

Claudio Lucchese[1], Raffaele Perego[1], Fabrizio Silvestri[1], Hossein Vahabi[1,2], and Rossano Venturini[1]

[1] ISTI-CNR, Pisa, Italy
[2] IMT, Lucca, Italy
name.surname@isti.cnr.it

Abstract. On-line photo sharing services allow users to share their touristic experiences. Tourists can publish photos of interesting locations or monuments visited, and they can also share comments, annotations, and even the GPS traces of their visits. By analyzing such data, it is possible to turn colorful photos into metadata-rich trajectories through the points of interest present in a city.

In this paper we propose a novel algorithm for the interactive generation of personalized recommendations of touristic places of interest based on the knowledge mined from photo albums and Wikipedia. The distinguishing features of our approach are multiple. First, the underlying recommendation model is built fully automatically in an unsupervised way and it can be easily extended with heterogeneous sources of information. Moreover, recommendations are personalized according to the places previously visited by the user. Finally, such personalized recommendations can be generated very efficiently even on-line from a mobile device.

1 Introduction

Designing an application for travel itinerary planning is a complex task, which requires to identify the so called Points of Interest (PoIs), to select a few of them according to user tastes and potential constrains (e.g. time), and finally to set them in a meaningful visiting order. Skilled and curious travelers typically consult several sources of information such as travel books, travel blogs, photo sharing sites and many others. The number of possible choices easily blows up, and makes it difficult to find the right blend of PoIs that best fits the interests of a particular user.

Many approaches have been proposed to automatically analyze the large amount of available information with the aim of discovering the most popular PoIs and routes. In this work we focus on media sharing sites such as Flickr[3]. The easiness and attractiveness of producing and sharing multimedia content through these sites motivate the exponential growth of the number of their users. Tourists are a typical example: while visiting a city, they took pictures of the most interesting places. These pictures are associated with a timestamp, often

[3] http://www.flickr.com

with geographic coordinates, and sometimes enriched with user-provided textual tags. A photo album can thus be considered an evidence of the route taken by a tourist while visiting a city. The goal of this work is to propose an effective PoI recommendations algorithm supporting interactive and personalized travel planning by exploiting the knowledge mined from Wikipedia and the billions of photos published in photo-sharing sites.

The raw data at hand is thus given by a few pages describing the most relevant PoIs in a city and by a large collection of images described by some features (e.g., the time at which the photo was taken, the user ID, the textual tags, the coordinates of the geographic location). These raw data are however very noisy. User-provided tags, when present, can be too general to identify a specific PoI (e.g. "Europe Tour 2011", "New York"), or irrelevant (e.g. "Me and Laura"), or wrong, or misspelled. The same holds for the geographic coordinates, since they can be missing or have different precision if provided by a GPS device or by the users, or they cannot help to discriminate between two very close PoIs.

In this work, we propose a novel algorithm for planning travel itineraries based on a recommendation model mined from such information sources. The model is based on a graph-based representation of the knowledge, and exploits random walk with restarts to select the most relevant PoIs for a particular user. Differently from previous proposals, our recommender system relies in fact on an initial set of PoIs to be used as *query places*. Query places are important because they represent contextual information identifying tourists interests. We carefully evaluate the system by casting the recommendation problem into a prediction problem, and by evaluating the ability of our recommendation algorithm to correctly guess the PoIs actually preferred by tourists which posted their albums on Flickr.

The paper is organized as follows. Section 2 discusses related works. Section 3 describes how to exploit users' photo and other Web data sources to identify the PoIs in a given region, and to map photos into such set of PoI. Eventually, users itineraries are transformed into a graph-based model, where a node corresponds to a PoI and edges are weighted with the expected probability of a user going from one extreme of the edge to the other. Section 4 presents our random walk with restart approach to provide personalized PoI recommendations given a set of already selected/visited PoIs. Section 5 shows experimental results on real world data relative to the cities of San Francisco, Glasgow and Florence that measure the effectiveness of our approach. Finally, Section 6 discusses possible extensions of this work and draws some final conclusions.

2 Related Work

Mean shift [3] has been proven to be an effective clustering technique for the identification of the set of PoIs P: by exploiting the geographic information associated with the input set of photos, it is possible to discover which are the most relevant PoIs [8, 4, 13, 9, 2]. A drawback of this approach is that an *observation scale* parameter must be properly set by the user, affecting the minimum size of a region that can be considered a PoI. In addition, not all the images might have geographic information, thus discarding some of the available data. When the PoIs P are identified by means of geographic clustering, each cluster of images must be assigned with a textual description which makes sense for the user, that means to identify the subject of each cluster. This is achieved by means of nearest neighbor matching with gazetteers, which provide geographic coordinates of relevant locations [9], or by text analysis conducted on image description and tags [13, 10, 14, 4]. These PoI naming techniques rely on the level of detail provided by gazetteers and by the consistency of user provided tags.

We propose here to determine the valid set of PoIs using Wikipedia as external knowledge base. The advantage of using Wikipedia is twofold. First it identifies a large number of PoIs in every city, even the less popular ones. Second, it provides additional structures information about the PoI, e.g. a subdivision in categories. After having identified the PoIs P and mapped each user image to a single PoI, the temporal sequence of images taken by each distinct user can be trivially translated into a trajectory joining the sequence of visited PoIs. These sequences are used to build a model of touristic routes in a city. To this purpose, the PoI sequences can be mined for frequent sequential patterns as in [13], to discover interesting visiting sequences. According to a collaborative filtering approach, the set of visited PoIs can be used to build user profiles, and therefore to leverage historical data of similar users [2]. A different approach is adopted in [8], where the user behavior is modeled by a mixture of a topic model, similar to collaborative filtering, and a Markov model where a user going from a PoI to another identifies a transition between two states. This mixed model tries to take into consideration correlated locations which do not necessarily occur consecutively in the visiting history of a user. A graph-based model is introduced by [5]. The authors actually model a set of PoIs with a clique where nodes are associated to PoIs and weighted with a reward and a visiting time, while edges are weighted with an estimate of the transit time between two PoIs. The reward associated with a PoI is derived by the number of users visiting it.

The devised model is then used to recommend relevant PoIs. The authors of [13] use a sort of reinforcement algorithm that ranks higher in the model the frequent sequential patterns generated by authoritative users and that include popular locations. A drawback of this approach is that recommendations are not personalized. In [8], the list of locations visited in the past by the user is used to build incrementally new trajectories that maximize their likelihood in the mixed topic-Markov model. The bests k routes satisfying a maximum time and distance constraints are returned. In [5] the problem of generating recommendations is reduced to the Orienteering problem, and a greedy algorithm is proposed to find the route between two locations providing the maximum reward within a given time budget.

Random walks based methods were successfully used in some related recommendation or graph problems. In [12], repeated random walks are executed on the similarity graph to reach object similar to the current item of interest. The authors of [6] proposed an random walk on the user social network to discover an item's rating from trusted users . In [7], RWR is used to recommend music tracks by leveraging an extended graph whose nodes represent users, as well as tags and song. Finally, in [1], it is used to predict the future outgoing edges of a node in an evolving graph.

3 A Graph-Based Model of Touristic Itineraries

The proposed recommender system exploits a graph model of the itineraries covered by tourists during their visit. Such model is built via a completely automatic process exploiting both photos from a photo sharing portal (in particular Flickr) and Wikipedia. The model identifies the PoIs in a given region, and measures there relatedness from a user perspective.

Definition 1 (Itinerary Graph). *An* Itinerary Graph $G = (V, E, w)$ *is an undirected weighted graph where each node in* V *corresponds to a PoI of a given region or city, and edge* $e = (u, v)$ *connects two PoIs if they are likely to belong to the same touristic itinerary, and* $w(u, v)$ *weights such probability.*

During their visit, tourists shoot photos of their preferred PoIs, and share them on the Web also enriching those photos with comments, tags, and other metadata. The process of recognizing the PoIs of a city given such set of photos is not trivial, since it requires to determine the landmarks depicted in any given image. This process is made even more difficult by the noise present in the data, such as wrong or irrelevant tags, approximate GPS coordinates, etc. Therefore detecting the set of PoIs by exploiting geographical clustering or visual recognition is very difficult and may introduce a significant amount of noise.

We solve the problem of PoI identification by resorting to Wikipedia. We identify be the smallest region that encloses a given city. We collect the georeferenced Wikipedia articles that fall within this region and consider the title of each articles as a PoIs name.

In the subsequent phase, we query Flickr to find the photos whose tags contain exactly the name of a PoI. For a given region, the number of results obtained with these queries is fewer with respect to other types of queries, e.g. spatial queries. False positive are however rare, since it is unlikely that a user adopted a very specific tag by chance or mistake. If an image is tagged with a Wikipedia article title, then it is very likely that the image renders the corresponding PoI. For each retrieved image, the user id and the corresponding PoI are sufficient to build an *itinerary graph* G for the given region. A node u is added to G for every PoI detected. An edge $e = (u, v)$ is added if there is at least one user that visited both u and v. We do not consider the timestamp of each photo, since we are not interested to the fact that a certain PoI has been visited immediately before or after another one. Our main objective is indeed that of establishing relations of mutual interest between PoIs independently of the time of their visits. The weight $w(e)$, $e = (u, v)$, is set equal to the number of different users that have the two PoIs u and v in their albums.

The resulting graph G models the co-visit frequency of any couple of PoIs. Such frequency is thus estimated on the basis of the actual user trajectories derived from the Flickr data. The graph G is used to estimate the probability that a user having visited a PoI u is also be interested in visiting the PoI v.

We further enrich the graph G by exploiting Wikipedia categories. Categories, indeed, provide strong signals about the correlation between two PoIs. PoIs may belong to the same class in some classification, or share the same architect or building period and so on. We derive another graph from Wikipedia, where the edge weights are given by the number of categories shared by two PoIs. Then, the two sets of edges are merged by equally weighting the Flickr-based and the Wikipedia-based contributions. In the experimental section, we show that the injection of Wikipedia-based relations, succeeds in automatically discover the topic the user is currently interested in, and in recommending new PoIs that belong to the same category.

In general, many other information sources or different features could be included in the model, thus extending the recommendation capabilities beyond the current analysis. A specifically tailored application, could also allow the user to choose between different perspectives, i.e. different ways of generating recommendations based on different information sources. Indeed, just basing on Flickr could harm the serendipity of the system.

4 RWR-based PoIs recommendation

In this section we propose a strategy for recommending PoIs given the itinerary graph $G = (V, E, w)$. We recall that G has a vertex for each PoI in the city. As described in Section 3, the edge $e = (u, v)$ belongs to E whenever a relation between the PoIs corresponding to u and v has been discovered (namely, the two PoIs are together in the album of at least a Flickr user or they share at least a category in Wikipedia). The weights $w(u, v) = w(v, u)$ is the number of these relations: the number of Flickr users and the number of shared Wikipedia categories.

Our recommendation algorithm assumes that the user has already visited, or that she has already showed interest to a set of PoIs, corresponding to a set of nodes $U \subset V$. Its aim is that of ranking the remaining PoIs in G with respect to the ones in U. The set U is used as a sort of user profile to personalize the generated recommendations. Ranking *all* the remaining PoIs is useful for various post-processing phases. For example, if the tourist is at her desk and she is planning a visit to the city, the system can simply show the top-k PoIs in the computed ranking, and support her in the interactive *selection and recommendation* of PoIs. On the other hand, if she is currently visiting the city, the ranked PoIs could be filtered by removing all the PoIs that are too far from her current position.

Given the graph G and the set of PoIs U which already attracted user's interest, our objective is that of scoring each node of $V \setminus U$ based on the level and the weight of its interconnection with nodes corresponding to PoIs in U. Our proposal relies on combining the results of Random Walks with Restart (RWR) (see [11] and references therein) started from each node in U in order to obtain the final ranking.

The graph G modeling the mutual relationships between the various PoIs is used to estimate the probability that a user having visited the PoI u is willing to visit the PoI v as well. This estimate is obtained by normalizing the weights of the given graph.

Definition 2 (Itinerary Transition Matrix). *Given the itinerary graph $G = (V, E, w)$ the* itinerary transition matrix $A \in \mathbb{R}^{|V|\times|V|}$ *estimates the conditional probability $P(v \mid u)$ for any given pair of nodes u,$v \in V$ by $A_{u,v} = w(u,v)/\sum_z w(u,z)$.*

The intuitive meaning of a random walk with restart from a node z is the following. A random walker starts visiting the graph G from node z. At each step, she is on a certain node u and has two possibilities: she can move to a neighbor of u or she can jump back to the initial node z. One of these two possibilities is chosen with probability α and $1-\alpha$ respectively, where $\alpha \in [0,1]$ is a real parameter. In the first case, the neighbor v is chosen with probability $A_{u,v}$. Note that A is not symmetric. The RWR from z is the steady-state probability r_z of this process. r_z is a vector of probabilities summing up to 1. The vector r_z is the solution of

$$r_z = \alpha r_z \times A + (1-\alpha)e_z$$

where A is the itinerary transition matrix derived from G, and vector e_z has all its entries set to 0 but the z-th which is 1.

The steady distributions r_z are computed for each PoI $z \in U$ and then merged by computing their Hadamard product in a single scoring vector r_U.

Definition 3 (PoI Score). *Given the itinerary transition matrix A resulting from the itinerary graph G, a seed set of PoIs corresponding to the set $U \in V$ induces a scoring of the other nodes of the graph, defined by $r_U(j) = \prod_{z\in U} r_z(j)$.*

The reason for resorting to the product of the entries instead of their sum is that we are more interested in discovering PoIs that are strongly related to most of the PoIs in U instead of PoIs that are highly related to just few of them [11].

Once r_U is computed, the recommender system suggests the k PoIs having the highest probabilities in r_U. As anticipated, the actual suggestion could be preceded by a preprocessing phase that filters out some of the PoIs (e.g., PoIs that are too far from the current position of the tourist), or that rearrange them in order to provide a suitable visiting path.

We finally observe that computing each vector r_z is a task too demanding to be performed at query time. Thus, in our solution, all the vectors are precomputed offline and stored in memory. The overall space occupancy is quadratic in the number of PoIs of the city. This is not problematic since even the most rich cities have at most a few thousands of PoIs. For the same reason, the final ranking vector r_U is computed efficiently: it passes through the computation of few (i.e., $|U|$) products of relatively small vectors.

	Florence	Glasgow	San Francisco
Number of PoIs	1,022	353	550
Images gathered from Flickr	124,223	176,981	937,389
Number of distinct albums (at least two photos)	2,919	1,971	4,411
Average distinct PoIs per album	3.71	4.97	3.61
Number of edges	131,238	25,486	39,372
Edges from Flickr	22,164	19,150	26,752
Edges from Wikipedia's Categories	111,778	8,644	16,038
Maximum (out)degree	415	103	263
Average (out)degree	121.86	72.20	71.59

Table 1. Statistics regarding the three datasets used in our experiments.

Since the ranking of the various PoIs is very efficient, our proposed algorithm can profitably be used in any interactive travel planning application running even on a mobile device. In this case, it is very easy to filter the ranked PoIs according to some external features (e.g. location, distance, reachability), or to incrementally build a route by recomputing the recommendations incrementally every time a user adds a new PoI to her route. Efficiency and extendibility are two distinguishing features of our approach.

5 Experimental Evaluation

First of all, we present in Table 1 some statistics regarding the datasets, i.e. the PoIs and the graphs obtained for Florence, Glasgow, and San Francisco. Fig. 1, instead, shows a list of the top-10 PoIs in each of the considered towns along with their normalized frequency in the datasets.

Evaluating the effectiveness of recommender systems is a difficult task as perceived quality is a subjective characteristic. To overcome to the lack of objective measurements, we cast our PoI recommendation problem into a PoI prediction problem and we evaluate the ability of our recommender system to correctly guess what a tourist visited in a town.

We consider thus the following problem: given a bunch of PoIs from a list of places in a given town actually visited by a tourist, the system must correctly guess what are the remaining favorite places the tourist visited in that town. As an example, suppose Alice during her tour of Barcelona visited the following five places: "*Sagrada Familia*", "*Parc Güell*", "*Casa Milà*", "*Casa Batlló*", and "*Picasso Museum*". Thus, when queried by using the first three PoIs of the above list the recommender should guess "*Casa Batlló*" and "*Picasso Museum*". The closer the number of correct guesses to the maximum possible, the better the quality of the recommender.

More formally, let V_i be the set of interesting PoIs for tourist i. We select a subset $U_i \subset V_i$ of size $\lfloor \frac{1}{2} |V_i| \rfloor$. We apply our algorithm to compute r_{U_i}, i.e. the vector containing the scores for each PoI in the town relative to the PoIs in U_i. Let $S_i@k$ be the set of top-k scoring PoIs according to r_{U_i}. The Normalized Precision at k (NP@k) is used to measure the precision of an algorithm in predicting the PoIs in V_i given the PoIs in U_i. NP@k is defined to be equal to $\frac{\sum_i |S_i@k \cap V_i|}{\sum_i \min\{k, |V_i \setminus U_i|\}}$.

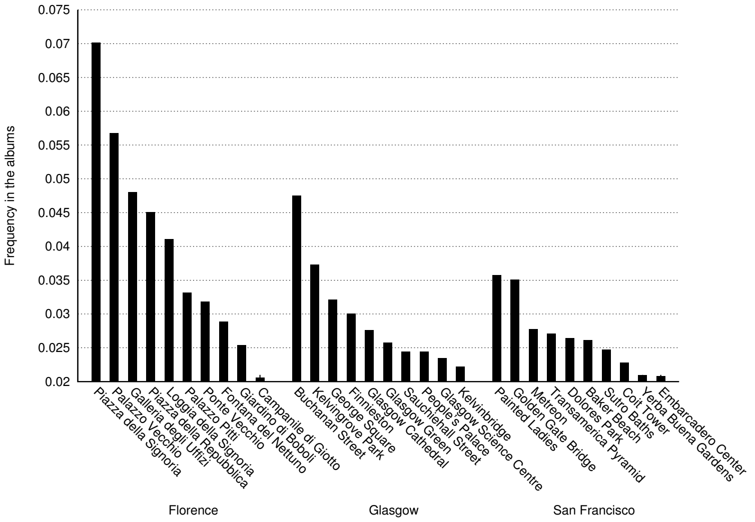

Fig. 1. Normalized frequency of the top-10 most frequent Pois in the Flickr photo albums for the three cities considered.

Basically, NP@k measures the overall number of suggestions correctly guessed normalized w.r.t. the maximum number of corrected recommendation possible when k recommendations are requested, i.e. $\sum_i \min\{k, |V_i \setminus U_i|\}$.

We tested our model using a 5-fold cross validation process. Models were built out of itineraries randomly chosen from those identified using the method described in Section 3. The remaining fifth is then used as a test set to evaluate NP@k values. Furthermore, since RWR uses a damping parameter α to decide restarts, we have evaluated what is the best value for α. We experimentally observed that the value α is to be considered independent of k. The values of NP@5 varying the parameter α are reported in Fig. 2. It is evident that small α values correspond to better results independently from the dataset on which the method is applied. For this reason RWR method results thereinafter are obtained with $\alpha = 0.2$

The baseline recommendation algorithm we adopt consists in suggesting independently from the subset $U_i \subset V_i$ the set of the top-k PoIs in the database, i.e. the k most visited PoIs of a given town. We refer to this strategy as "*Touristic Guide*". Notice that for tourism related applications, this is a hard-to-beat baseline. In fact, the most visited PoIs are by far the most popular ones. Indeed, from an estimate we made using our datasets about 20% of PoIs are visited by at least 80% of tourists.

In Fig. 3 we report NP@k when k ranges from 1 to 5. We tested both random and sorted samplings for query selection. Random query selection consists of choosing query PoIs randomly for each tourist. Furthermore, to better simulate the behavior of a tourist willing to visit a city, we also tested a sorted selection

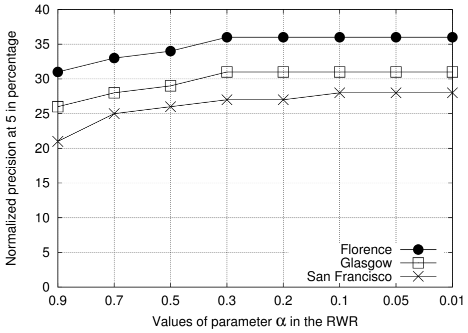

Fig. 2. NP@5 varying the α parameter in RWR.

that samples most popular $\frac{1}{2}$ PoIs in each V_i. The idea is that we want to measure how good our system is at recommending useful, yet not popular, PoIs.

For all the values of k, our RWR method outperforms sensibly the Touristic Guide strategy both when queries are randomly selected and when the most popular ones are chosen. In the case of queries made up of popular PoIs NP@k are lower than the random query selection. This is expected, indeed, given that our recommenders are not allowed to propose places already visited and that, in this case, popular destinations are already used as queries and cannot be suggested.

A possible explanation for RWR superiority is that it is able to recommend places that are related with those already visited. Instead, the Touristic Guide strategy is oblivious to the history of a user and this may degrade recommendation quality. The same observation holds also in the case of randomly selected query PoIs.

To conclude, we present some examples of suggestions computed by means of our algorithm. The aim of the following examples is that of showing the behavior of our system on PoIs of Florence, Glasgow, and San Francisco.

In the first example the set of starting PoIs U contains two of the most important PoIs of Florence: *Palazzo Vecchio* and *Piazza della Signoria*. The top-10 PoIs ranked by our recommender are shown in the Table 2(a).

Without any doubt these 10 PoIs are among the most important PoIs in Florence. In presence of very popular PoIs in U, our system responds by producing a ranking that has other very famous PoIs on its top. These are conditions where the edges gathered from Flickr come into play. Most of the tourists perform, in fact, tours of the city by mainly visiting its most important PoIs. Thus, since many albums in Flickr contains all these PoIs together, our graph has a large component that connects all of them. This component tends to increase the ranking probabilities of these PoIs when some of them belong to U.

For the second example we selected the following, less famous, four PoIs: *La Specola, Museo Fiorentino di Preistoria, Museo Horne* and *Bargello.* These

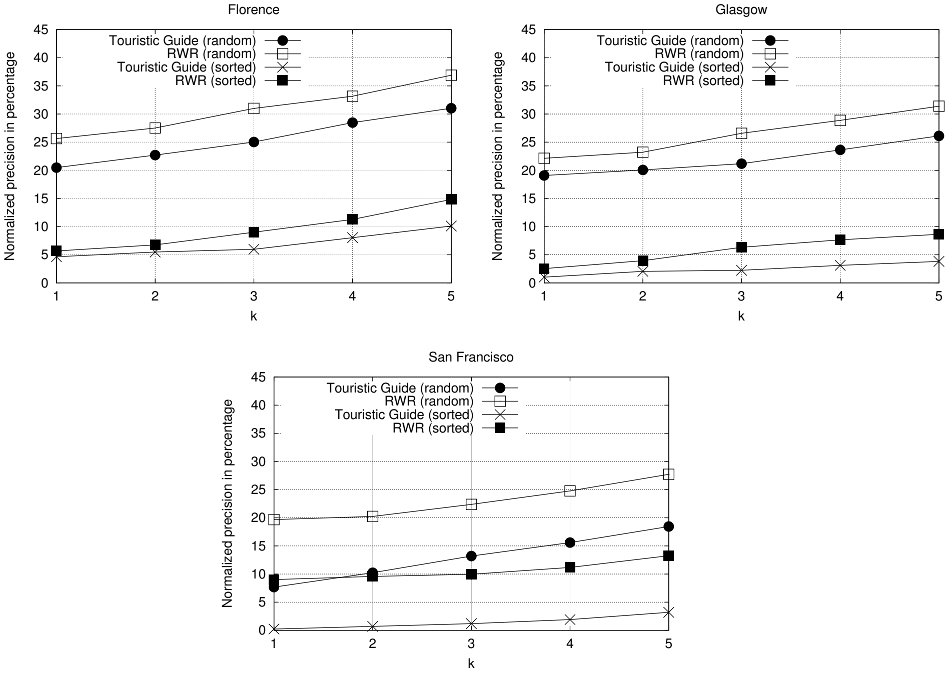

Fig. 3. Normalized Precision NP@k as a function of *k*.

are all museums in Florence. The top-10 PoIs ranked by our recommender are reported in the Table 2(b).

We observe that the top-10 ranked PoIs can be classified as museums. This kind of response is mainly due to the structure gathered from Wikipedia. As we already pointed out, we are able to relate together PoIs that are semantically similar by exploiting categories of Wikipedia. We also observe that these museums are presented in a order that reflects their relative importance. For example, *Uffizi* is probably the most important museum in Florence. This second effect is again a consequence of the edges extracted from Flickr.

The third example referred to the city of Glasgow shows how our system is able to adapt itself to the expected needs of the user. We start from four PoIs: *Clyde Tunnel, Govan Subway Station, Hillhead Subway Station* and *Renfrew Airport*. We have a tunnel that connects two parts of the city, two subways stations and the Glasgow's domestic airport. The returned PoIs are reported in the Table 2(c).

In this case all the top-10 PoIs identified by our model are highly related to transportation within the city of Glasgow. Among these results, we can find a airport, a heliport, a seaplane terminal, a bus station and a few subway stations. Even in this example, correlations learned from Wikipedia help our model

Starting PoIs in U
Palazzo Vecchio
Piazza della Signoria

Top-10 ranked PoIs

PoI	Probability
Ponte Vecchio	$5.9 \cdot e^{-4}$
Piazzale Michelangelo	$2.1 \cdot e^{-4}$
Palazzo Pitti	$1.9 \cdot e^{-4}$
Giotto's Campanile	$6.8 \cdot e^{-5}$
Boboli Gardens	$4.9 \cdot e^{-5}$
Loggia dei Lanzi	$4.6 \cdot e^{-5}$
Piazza Santa Croce	$4.2 \cdot e^{-5}$
Uffizi	$4.1 \cdot e^{-5}$
Basilica of Santa Croce	$3.9 \cdot e^{-5}$
Ponte alle Grazie	$3.4 \cdot e^{-5}$

a)

Starting PoIs in U
La Specola
Museo Fiorentino di Preistoria
Museo Horne
Bargello

Top-10 ranked PoIs

PoI	Probability
Uffizi	$1.4 \cdot e^{-10}$
Giotto's Campanile	$1.2 \cdot e^{-10}$
Palazzo Medici Riccardi	$9.8 \cdot e^{-11}$
Vasari Corridor	$7.4 \cdot e^{-11}$
Medici Chapel	$6.5 \cdot e^{-11}$
Basilica of Santa Croce	$5.3 \cdot e^{-11}$
San Marco's National Museum	$1.3 \cdot e^{-11}$
Dante Alighieri's House	$9.6 \cdot e^{-12}$
Modern Art Gallery	$9.3 \cdot e^{-12}$
Museo Stibbert	$8.0 \cdot e^{-12}$

b)

Starting PoIs in U
Clyde Tunnel
Govan Subway Station
Hillhead Subway Station
Renfrew Airport

Top-10 ranked PoIs

PoI	Probability
Glasgow International Airport	$1.2 \cdot e^{-8}$
Buchanan Street Subway Station	$4.2 \cdot e^{-9}$
Kelvinbridge	$6.8 \cdot e^{-10}$
Glasgow Seaplane Terminal	$2.4 \cdot e^{-10}$
St Enoch Subway Station	$2.0 \cdot e^{-10}$
Glasgow City Heliport	$2.0 \cdot e^{-10}$
Buchanan Bus Station	$9.5 \cdot e^{-11}$
Ibrox Subway Station	$9.5 \cdot e^{-11}$
Kelvinhall Subway Station	$8.3 \cdot e^{-11}$
Cowcaddens Subway Station	$9.5 \cdot e^{-12}$

c)

Starting PoIs in U
Golden Gate Theatre
San Francisco Conservatory of Music

Top-10 ranked PoIs

PoI	Probability
War Memorial Opera House	$1.1 \cdot e^{-5}$
Dolores Park	$1.0 \cdot e^{-5}$
Castro Theatre	$8.1 \cdot e^{-6}$
Yerba Buena Gardens	$7.8 \cdot e^{-6}$
Embarcadero Center	$7.3 \cdot e^{-6}$
Metreon	$6.3 \cdot e^{-6}$
Golden Gate Bridge	$5.5 \cdot e^{-6}$
Pacific-union Club	$4.2 \cdot e^{-6}$
Lake Merritt	$4.1 \cdot e^{-6}$
American Conservatory Theater	$3.9 \cdot e^{-6}$

d)

Table 2. PoI recommendations in Florence, Glasgow, and San Francisco.

to identify common aspects that relate PoIs in U. Finally, relative importance learned from Flicker is fundamental for ranking equally correlated PoIs. For example, we observe that the highest ranked subway station, *Buchanan Street Subway Station*, is the most central and busy station on the subway of Glasgow.

The last example is for the city of San Francisco. We start from the two PoIs: *Golden Gate Theatre* and *San Francisco Conservatory of Music*. The top-10 PoIs ranked by our recommender are shown in the Table 2(d). Even in this example, we have the same phenomenon observed in the previous ones: PoIs related to theaters, music, and culture in general are placed among the first positions.

6 Future work and conclusions

We believe that there could be many interesting ways to further improve our PoIs recommender system. An important part of its effectiveness depends on the quality of the relations between PoIs which are inferred and weighted by resorting to Flickr and Wikipedia. Thus, it is worth to try to enrich the graph by extracting relations from other heterogeneous sources of information (e.g., TripAdvisor, Lonely Planet, and so on). For example, we could exploit also the hierarchy of categories present in Wikipedia: a relation could have a boost whenever it is obtained from a very specific category. Moreover, more attention should be posed on the weighting phase. In Section 3 we implicitly assumed that all the relations have the same importance. These aspects should be further investigated and other signals and other weighting schema exploited.

References

1. L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM'11*, pages 635–644, 2011.
2. M. Clements, P. Serdyukov, A. P. de Vries, and M. J.T. Reinders. Using flickr geotags to predict user travel behaviour. In *SIGIR'10: Proceedings of the 33th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, ACM, July 2010.
3. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.
4. D.J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *WWW*, pages 761–770. ACM, 2009.
5. M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *HT*, pages 35–44, 2010.
6. M. Jamali and M. Ester. TrustWalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 397–406. ACM, 2009.
7. I. Konstas, V. Stathopoulos, and J. M Jose. On social networks and collaborative recommendation. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval SIGIR 09*, 8(3):195, 2009.
8. T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *CIKM*, 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. *MM*, pages 143–152, 2010.
10. T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR*, pages 103–110, 2007.
11. H. Tong and C. Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *KDD*, pages 404–413, 2006.
12. H. Yildirim and M. S Krishnamoorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. *Proceedings of the 2008 ACM conference on Recommender systems RecSys 08*, page 131, 2008.
13. Z Yin, L Cao, J Han, and J Luo T Huang. Diversified trajectory pattern ranking in geo-tagged social media. *SDM*, 2011.

14. Y Zheng, M. Zhao, Y. Song, H. Adam, U., A. Bissacco, F. Brucher, T. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092, 2009.