Personal and Ubiquitous Computing (2019) 23:199–214
https://doi.org/10.1007/s00779-019-01218-z

ORIGINAL ARTICLE

Point of interest recommendation based on social and linked open data

Giuseppe Sansonetti[1]

Received: 5 July 2018 / Accepted: 28 March 2019 / Published online: 17 April 2019

Abstract
Location-based services (LBSs) are part of our daily lives due to the huge spread of mobile devices. Such services enable us to access relevant and up-to-date information about our current surroundings at any time and everywhere. The adoption of a data-driven semantic layer coexisting with the traditional Web could help further improve LBSs, allowing them to overcome the barriers imposed by closed databases that do not take advantage of the large amount of public data available on the Internet. In this article, we propose a personalized recommender system of points of interest (POIs) located near the user's current position, which makes use of the gold mine represented by linked open data (LOD). The target user profile is constructed and updated using two differente sources of feedback. The former is obtained by analyzing her activity on social media (i.e., Facebook). The latter is attained by inviting the user to express her interests and preferences as ratings of a sample of selected images representing specific categories of POIs. Experimental tests performed on real users allowed us to verify the good performance in terms of perceived accuracy and normalized discounted cumulative gain (NDCG). Statistical tests also enabled us to verify the significance of all the obtained results.

Keywords Location-based services · Recommender systems · Social media · Linked open data

1 Introduction

The amount of content available on the Web is constantly growing and, with it, the difficulties in identifying the relevant information during a search. A user has to be able not only to define her interests clearly, but also to know how to manage the numerous sources of information available. This experience often causes a sense of overwhelming that can discourage the user well before she can achieve the desired results. The introduction of efficient Information Retrieval techniques [1, 2] and reliable recommender systems (RSs) [3, 4] by the major web content providers has partly mitigated those issues. Although inexperienced users may not realize it, these techniques are now adopted by most web platforms, thus largely affecting users' activity: from the choice of books to purchase on Amazon[1], to the next movie to watch on Netflix[2]. Suggestions and content personalization go hand in hand with users' satisfaction and, therefore, with merchants' profits. The introduction of this kind of systems into web applications has hence become a consolidated practice, perhaps even necessary. Despite the advancement of the technologies aforementioned, the main developers of these systems have proved rather reluctant to exploit semantic techniques. Those techniques have not only been extensively studied, but their practical application to the World Wide Web has already been proposed and defined since its beginnings in the 1990s. The adoption of a data-driven semantic layer coexisting with the traditional Web could contribute significantly to the improvement of most intelligent Internet systems. For instance, this could allow us to overcome the barriers imposed by closed databases that do not exploit the large amount of public data available on the Web [5, 6].

The study presented in this article concerns the application of linked open data (LOD) to intelligent systems for the Web in order to increase their performance. Moreover, some techniques for modeling the user profile through the information extracted from social media are investigated. These

[1] https://www.amazon.com/

✉ Giuseppe Sansonetti
gsansone@dia.uniroma3.it

[1] Department of Engineering, Roma Tre University, Rome, Italy

[2] https://www.netflix.com/

tools are used to design a system able to provide the target user with personalized suggestions related to geolocalized points of interest (POIs) nearby her current position. Hence, our overall aim is to answer the following research question: *can the combination of social and linked open data bring benefits to the domain of location-based recommender systems?*

The rest of this article is structured as follows. In Section 2, we illustrate some works related to the system proposed herein. The recommendation problem is formulated in Section 3. In Section 4, the advanced RS is introduced, focusing on its requirements and functionalities. The experimental tests performed for evaluating the system performance and the results achieved are reported in Section 5. Finally, we draw our conclusions and outline some possible future developments of this work in Section 6.

2 Related work

In this paper, we propose a personalized recommender of points of interest based on social and linked open data. This section describes some systems in the research literature, which offered useful hints for the approach presented here.

As for POI recommendation, many useful and efficient systems have been proposed in the literature [7, 8]. Among these, the authors of [9] provide a twofold contribution. In order to better characterize the target user's interests, they first consider a preference model based not only on her check-ins, but also comments on venues, processed through text-based sentiment analysis techniques. The authors then propose a matrix factorization approach (called location-based social matrix factorization, LBSMF) enhanced to include the effects of social influence and venue similarity in the location recommendation algorithm.

In [10], the matrix factorization approach is further extended for considering the semantic attitudes, that is, sentiment, volume, and objectivity, extracted from user-generated content. Potential temporal alterations of users' attitudes are also taken into consideration in the proposed model.

The SEAL (Sentiment-EnhAnced Location search) system proposed in [11] is a fine-grained preference-aware location search framework that exploits the information in the content generated by users on LBSNs. This system resorts to a factorization technique based on a three-way tensor to consider positive and negative user's preferences in the process of personalized location ranking.

The authors of [12] present a spatial temporal activity preference (STAP) model to address the problem of the high dimensionality and sparsity of the data to be handled. This model considers the spatial and temporal features of users' activities separately and employs tensor factorization techniques to extract their preferences from check-ins. The experiments performed on real data from two popular LBSNs allowed the authors to show the better performance of their model than those of other state-of-the-art approaches.

Also the recommender system proposed in [13] is able to take into account how user's interests evolve over time. The basic idea underlying such an approach, named bag-of-signals, is to model each potential user's interest as a signal.

As for LOD technologies, they have found various uses not only in experimental systems, designed with the purpose of exploring the potential of the Semantic Web, but also in practical applications. Currently, it is possible to identify three types of LOD-based applications [14]: browsers, search engines, and specific applications. In this scenario, we focus on the latter, particularly on RSs that take advantage of LOD. Many examples of RSs are presented in the literature, but those that integrate semantic techniques with linked open data are still a minority. What motivates this caution towards LOD is probably given by their sectoral characteristic: if for some domains of interest there are updated and rich datasets, the same cannot be claimed for others. This phenomenon is likely to gradually decrease, as the LOD cloud is constantly expanding. Below, we report a quick overview of some representative studies conducted in this regard. In [15], the author proposes several theoretical methods for assessing the semantic distance between two entities, which can be seen as a measure of how closely related they are. This theory has been taken up in several works to develop recommender systems.

The authors of [16] advance the use of semantic relations of objects positively evaluated in the past by the active user to suggest new relevant elements to her. The relationship between the two objects allows for the explanation of the recommendation as well. Furthermore, the authors also analyze the problem of recognizing semantic relations actually relevant to the purpose of an application. This aspect becomes even more significant when the relationships involve different ontologies. Through a series of tests conducted on real users, the authors were able to identify the semantic patterns leading to the most interesting results and exclude those that are not relevant.

A similar concept is studied in [17] for content-based filters. Whereas in traditional systems the similarity between two objects is calculated based on their descriptions, the authors exploit LOD to express the concept of similarity as the amount of shared similar information. In other terms, two resources can be considered similar if in the RDF graph that constitutes LOD,

– they are directly connected through a predicate;

– they are the subjects of two triples with the same predicate and object;
– they are the objects of two triples with the same subject and predicate.

Based on this theory, the RDF database is represented as a three-dimensional matrix where each section refers to a property of the ontology and represents its adjacency matrix. A matrix element has a non-null value if there is a property that relates the subject (row) to the object (column). In this way, by fixing the first dimension (the value for a predicate), it is possible to assess the similarity between two entities by comparing the single vectors of the matrix by means of metrics such as the cosine similarity. The authors model the user profile through the entities that she liked in the past in order to find the most similar candidates to suggest to her.

In [18], the authors propose the use of an alternative user modeling technique and its integration with LOD to extract relevant content during a museum visit. The user's interests and the POI characteristics are defined by means of tag sets and enriched in turn through LOD. By measuring the semantic distance between tags of the user profile and tags of the available POIs, it is possible to determine the best candidate for the suggestion. The presentation of recommended POIs is further adapted to the user's personality inferred while observing her itinerary during the visit. Less interested visitors are suggested more interactive POIs to increase their involvement. Conversely, more curious visitors are guided towards POIs that require to dwell on documents or other forms of media.

However, developing this kind of system poses some challenges in the data collection phase. Among those, the most harmful problems are *cold-start* and *data sparsity*. As a result, Heitmann and Hayes [5] advance an approach that leverages LOD to mitigate such phenomena along with the most portable recommendation algorithms, so that they can be applied to any domain of interest. The authors identify three components in traditional RSs: (i) background data retained by the system a priori, (ii) input data about objects and users, and (iii) recommendation algorithm. Their system includes two additional layers: an interface with data for extracting LOD from external endpoints and an integration service for converting them into a homogeneous format and blending them with background data. The latter, represented as a traditional user-item matrix, allow for the application of the classic collaborative algorithms. Through the illustrated method, it is possible to propose new items or offer suggestions to a new user even if the RS does not have any information on their characteristics, supplying this lack of knowledge by means of LOD available on the Web.

A hybrid recommendation model based on semantic concepts is discussed in [19]. Instead of evaluating the similarity between two users on a global scale, where many facets could be lost, the authors propose a distinction of their interests in different layers. The layers are represented by means of preference vectors for the various concepts of an ontology, the use of which produces a less ambiguous model than the one that would be by using single objects or keywords. The ontological basis also allows the system to identify the relationships between individual concepts thanks to well-defined semantic propositions. Therefore, multiple values of similarity spread over various subsets of interests are determined between two users, which allow real clusters centered on categories of objects to be defined.

In [20, 21], the authors present a system that combines the content-based and collaborative approaches with techniques based on social and linked open data. By means of ideas borrowed from the Semantic Web, the information extracted from Facebook is converted into a structured form that allows it to be used in synergy with LOD within a recommender of cultural heritage venues. Users' posts are examined to determine relevant entities and the extracted information is represented as a graph of concepts linked through semantic relationships. Unlike the previous approaches, this one exploits information regarding the user's social graph for giving more relevance to POIs visited by her friends in the recommendation process.

All the reported studies have been tested and evaluated through the classic metrics used in this research field, showing not only that the use of LOD for developing RSs is possible but also that it allows them to attain performance comparable to those of current commercial systems. Nevertheless, systems combining social and linked open data are still a minority.

3 Problem formulation

In this section, we provide the definition of the recommendation problem of POIs located nearby the active user's current position.

Let $\mathbb{U} = \{u_1, \cdots, u_N\}$ represent the set of N users with a valid account on social media. For each user $u_i \in \mathbb{U}$, we build her user profile P_{u_i} by collecting and analyzing her feedbacks expressed as *like* and clicks on pictures (see Section 4.1.1). Let $\mathbb{L} = \{l_1, \cdots, l_M\}$ represent the set of M candidate POIs, namely, the POIs extracted from LOD and located in the user's surroundings. The last point means POIs placed within a circle centered in the user's current position and having a radius r, whose value is set by her. Under those settings, the problem can be formulated as follows: define the function f

$$f : \mathbb{U} \times \mathbb{L} \rightarrow [0, 1] \tag{1}$$

such that, given a target user u_i represented through her profile P_{u_i} and a set of candidate POIs, f expresses the recommendation score of the candidate POI l_j for the target user u_i. After obtaining the recommendation scores for every candidate POI, we rank all the POIs according to their scores and return them as a top-k recommendation list.

4 The proposed system

This section describes the system implementation, motivating the various choices made and presenting the problems encountered. The main system modules, the techniques used, and the algorithms implemented for its realization are outlined. Furthermore, the graphical interface is shown as well as the modalities in which the user can interact with the system.

4.1 Architecture

The architecture of the system can be divided into four main modules, represented in Fig. 1:

1. User profile extraction;
2. POI extraction from LOD;
3. Recommendation process;
4. Presentation and evaluation of results.

4.1.1 User profile extraction

This section presents how the user profile needed for customizing the recommendation process is built. To this aim, we take into account two different sources of user feedback: her activity on social media and her clicks on a sample of images representing specific categories of POIs.

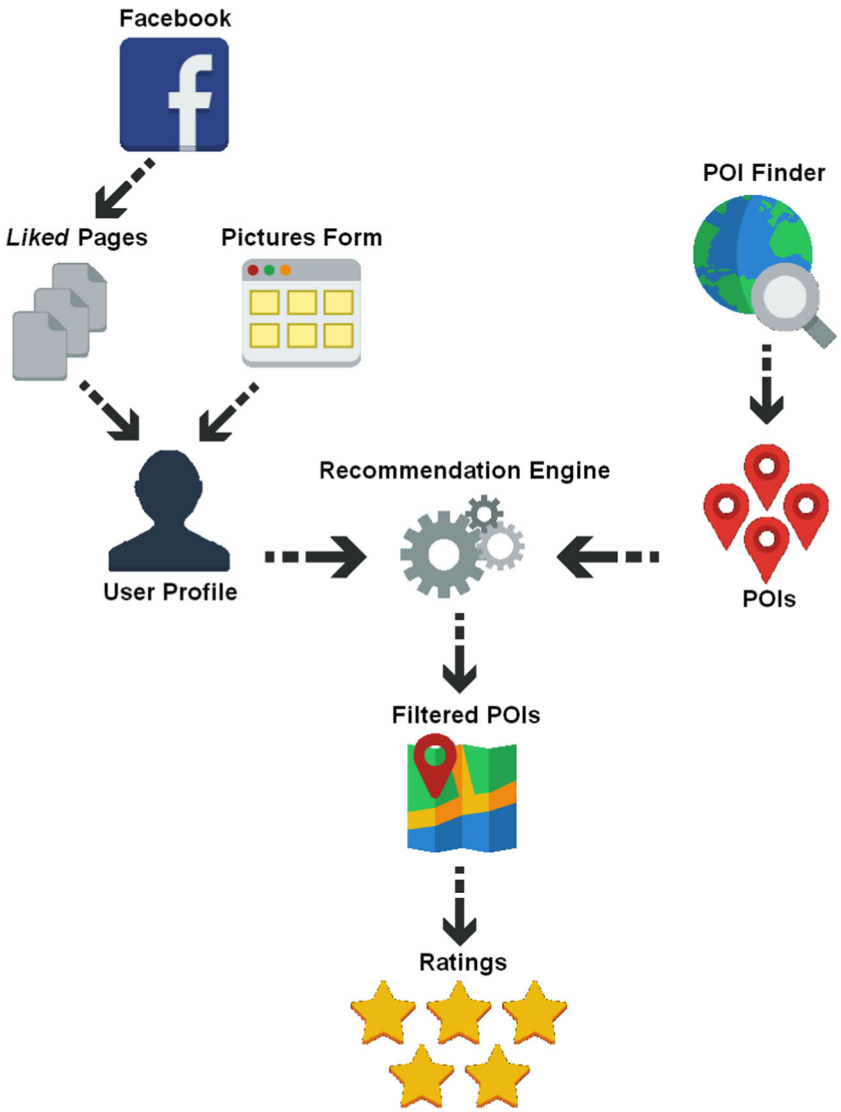

Fig. 1 System architecture

Activity on social media As first source of user feedback, we pointed towards something already available on the Web for most users: their activity on social media. More specifically, we considered Facebook[3], one of the most popular social media. As of January 2018, Facebook is used by 42% of the world's population, with peaks of 70% in North America and 66% in Northern Europe.[4] Through a user's Facebook profile, the system leverages the related API[5] to extract her demographic information (i.e., age, gender, and profession), and estimate her interests and preferences by analyzing the pages tagged by her with a *like*. However, the simple extraction of the user's *like* is not sufficient. It is necessary to process them so that they can be compared directly with POIs. Ideally, we would like to obtain the equivalent of the Facebook page in the same domain as the points of interest, namely, DBpedia[6]. However, the retrieved data often presents three major problems:

1. It has a large amount of noise, that is, pages that cannot be associated with well-defined entities (mainly personal blogs and pages that publish content not relevant to the system);
2. It needs a disambiguation process for determining the entity to which it refers (e.g., the "house" page could indicate the music genre, the TV series, the movie, and more);
3. It can contain multiple separate pages that refer to the same concept (e.g., two separate pages, one official and one not, dedicated to the same celebrity).

To cope with such problems, a mapping between the categories of Facebook pages (i.e., seven macro-categories with numerous sub-categories) and the classes of the DBpedia ontology has been provided (see Table 1). Through this mapping, it is possible to use the name of the Facebook page to explore DBpedia for extracting all the entities with the same name. A filter is then applied for removing the entities that do not belong to the class obtained by mapping the category of the Facebook page to the DBpedia ontology. To this aim, we perform a parametric SPARQL query that solves the three problems aforementioned:

– Most of the noise pages are filtered out, because their names hardly find a match on DBpedia and some categories of pages are a priori removed (i.e, blogs, podcasts, and others);
– There is no ambiguity because the category of the Facebook page allows us to obtain an exact match;

[3] https://www.facebook.com/

[4] https://www.statista.com/statistics/269615/social-network-penetration-by-region/

[5] https://developers.facebook.com/

[6] http://wiki.dbpedia.org/

Table 1 Mapping between the Facebook categories and the DBpedia ontology

Facebook category	DBpedia ontology
Art	Artwork
Movie	Film
Movie Character	FictionalCharacter
Music	MusicalWork
Musician/Band	MusicalArtist
Personal Blog	
Podcast	
TV Show	TelevisionShow
Video Game	VideoGame
Website	Website
...	...

– Duplicates are removed by the SPARQL query requiring only distinct results.

In this way, it is possible to univocally associate each user's *like* to the corresponding entity of DBpedia (e.g., see Fig. 2).

Clicks on a pictures form As second source of user feedback, we ask the user herself to provide the system with her interests and preferences. Instead of forcing the user to compile a verbose and tiresome multiple choice questionnaire, we preferred to adopt an easier and faster system. More specifically, the user is presented with a gallery of images depicting particular categories of places, so she can select the ones more interesting to her by simply clicking on them (see Section 4.3 for more details). Those images were selected from those on the Flickr[7] image sharing website based on the tags assigned by users. In particular, we chose those with the highest possible agreement among taggers. With the user not aware of the details below, each image is associated with one of the following ten categories based on the model adopted in the location-based social network (LBSN) Foursquare[8] used in this system to classify POIs:

1. *Arts&Entertainment*
2. *College&University*
3. *Event*
4. *Food*
5. *NightlifeSpot*
6. *Outdoors&Recreation*
7. *Professional&OtherPlaces*
8. *Residence*
9. *Shop&Service*
10. *Travel&Transport*

[7] https://www.flickr.com/

[8] https://developer.foursquare.com/docs/resources/categories

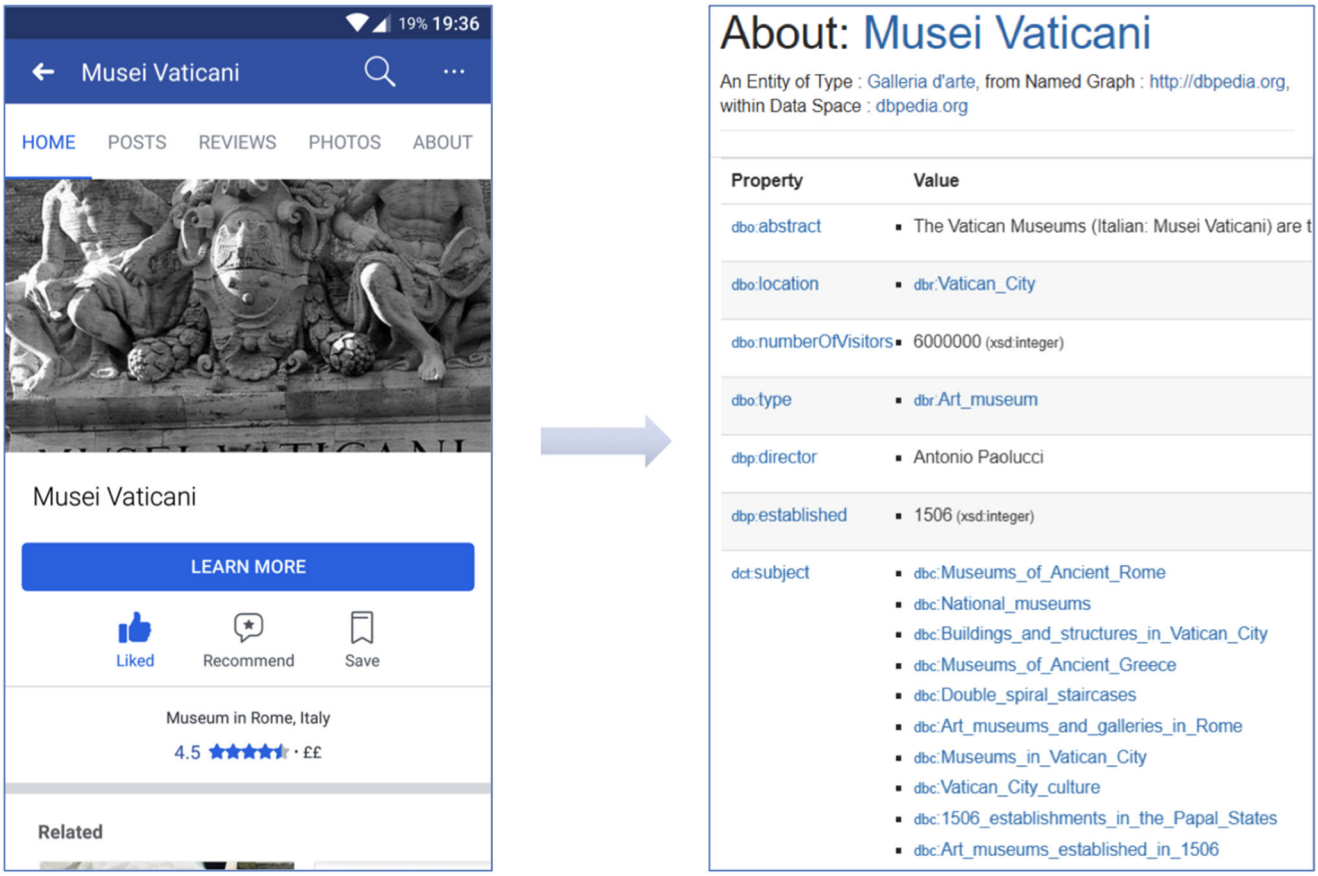

Fig. 2 Association of a user's *like* to the related entity of DBpedia

Each category is represented by more than one image and has been chosen to present them in random order and without explicit labels. In this way, the user's choice depends exclusively on the sentiment evoked by the image. At the end of the selection, the system generates a vector of preferences where each element represents the number of images selected for a specific category (see Fig. 3). For example, if the user selects two images related to the *Event* category, one related to *Arts&Entertainment* and three related to *Outdoors&Recreation*, the vector associated with her would be the following:

$$< Arts\&Entertainment, College\&University, Event, Food, NightlifeSpot, Outdoors\&Recreation, Professional\&OtherPlaces, Residence, Shop\&Service, Travel\&Transport >= =< 1, 0, 2, 0, 0, 3, 0, 0, 0, 0 >$$

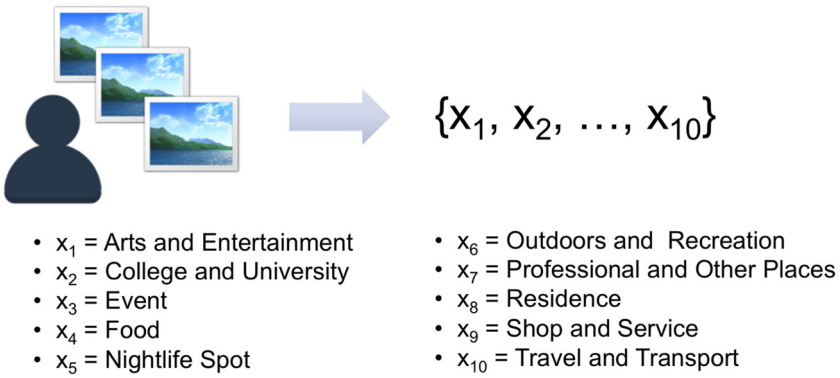

Fig. 3 User profile generation

By normalizing the values (i.e., dividing each component by the sum of all the components), we obtain a new vector of components between 0 and 1, whose sum is equal to 1.

$< 0.167, 0, 0.333, 0, 0, 0.500, 0, 0, 0, 0 >$

The higher the value associated with a certain category, the higher the user's interest in it. With the vector above, POIs associated with the *Outdoors&Recreation* category would be considered the most similar to the user's preferences. Afterwards, we would have *Event*, *Arts&Entertainment*, and then all the others with a value equal to 0.

4.1.2 POI extraction from LOD

To suggest a set of POIs potentially relevant to the active user, the system has to be able to submit a query to a LOD endpoint by providing filters based on the selected search area. In this way, it can retrieve all the information useful to determine the relevance of each POI. Since this operation requires the exchange of data through the network with external systems, it is essential to optimize the way in which it is performed for reducing the waiting time as much as possible. For this purpose, the POI Finder module shown in Fig. 1 is responsible for sending a properly formulated SPARQL query to the DBpedia endpoint and retrieving the data available in RDF format. The data is then converted as table by the server so as to be more easily manipulated in the subsequent phases, and temporarily stored in memory. In particular, the following information is extracted for all DBpedia objects that are geolocated in the area at hand:

- Object URI
- Descriptive label
- Latitude and longitude
- Image link (if available)
- Wikipedia page link
- Abstract
- Type

Once this information is retained into memory, the system takes care of converting the `rdf:type` property from the DBpedia ontology to the Foursquare categories used by the system and calculating the distance of the POI from the center of the search area. As well as the Facebook page categories, the types of DBpedia relevant to this application have also been mapped to the ten categories adopted by the system. An example of output of the POI Finder module is shown in Table 2.

Table 2 Example of POI Finder module output

URI	Label	Latitude	Longitude	Image	Wikipedia	Distance	Type	
http://dbpedia.org/...	Temple of Diana (Rome)	188.08	92.32	http://commons.wikimedia.org/wiki/...	http://en.wikipedia.org/wiki/...	The Temple of Diana in ancient Rome was a Roman temple	01.03	Professional and Other Places, Outdoors and Recreation
http://dbpedia.org/...	Temple of Minerva (Aventine)	188.06	92.26	http://commons.wikimedia.org/wiki/...	http://en.wikipedia.org/wiki/...	The Temple of Minerva was a temple on the summit of the Aventine Hill in Rome	01.07	Outdoors and Recreation
http://dbpedia.org/...	Teatro Argentina	190.16	91.22	http://commons.wikimedia.org/wiki/...	http://en.wikipedia.org/wiki/...	The Teatro Argentina is an opera house and theatre located in Largo di Torre Argentina	01.25	Arts and Entertainment
http://dbpedia.org/...	Cavour (Rome Metro)	189.5	94.16	http://commons.wikimedia.org/wiki/...	http://en.wikipedia.org/wiki/...	Cavour is a station on Line B of the Rome Metro	0.58	Travel and Transport

4.1.3 Recommendation process

Candidate POIs are those available in the user's surroundings, that is, within a r radius whose value is chosen by the user herself. Once such POIs have been found, the most pertinent ones have to be selected through the application of the recommendation algorithms. The six modules described below are designed for assigning a relevance value between 0 and 1 to each POI based on a different function f defined in Section 3. Through this value, the list of POIs can be sorted from the most relevant to the least relevant. To avoid overwhelming the user with an excessive number of results, we chose to select only a small number of POIs from the head of the list.

Random selection (R) As the name itself suggests, this module assigns random values as recommendation scores for POIs. Although used alone it does not constitute a reliable system, it is deployed as a baseline. It is expected that any recommendation algorithm will perform at least as good as the baseline.

Popularity-based selection (P) This module returns the list of candidate POIs sorted in descending order based on their popularity value, regardless of the target user profile. The popularity value is given by the number of check-ins gathered on the LBSN Foursquare, which can be obtained through an appropriate query. For this purpose, it is necessary to map the POI candidates extracted from LOD to the POIs available in Foursquare using the modalities outlined above.

Content-based selection (C) Content-based selection exploits the user feedback described in Section 4.1.1. Since the categories have been extracted for each POI, it is possible to define a vector in the same way as seen for the user profile but, unlike it, the components associated with the categories are Boolean values: 1 if the POI belongs to the category, 0 otherwise. Furthermore, this vector is not normalized. This choice allows us to run the scalar product between the vector $\overrightarrow{l_j}$ representing the POI and the vector $\overrightarrow{P_{u_j}}$ representing the profile of the user u_j:

$$\begin{aligned}\overrightarrow{l_j} \cdot \overrightarrow{P_{u_j}} &= < x_1, x_2, ..., x_n > \cdot < y_1, y_2, ..., y_n > \\ &= x_1 y_1 + x_2 y_2 + ... + x_n y_n \end{aligned} \quad (2)$$

thus obtaining a numerical value between 0 and 1. Such a value expresses the similarity between the two entities. The categories selected more times by the user are preferred over the others, so contributing more to the final score attributed to the POI.

Tag-based selection (T) This module takes advantage of the properties of DBpedia objects to compare them. In theory, two distinct POIs are considered the more similar, the more numerous are the `dct:subject` properties in common. In practice, looking for direct correspondences between the subjects of the items involved may not always produce satisfactory results because some POIs show a very reduced list. To compensate this sparsity, an annotation service[9] is used, which enables the system to extract references to DBpedia with confidence values between 0 and 1 from an input text. In this way, it is possible to express a given POI according to the tags extracted from the `dbo:abstract` property of its corresponding Wikipedia page. All the tags that fall under a confidence threshold established during the experimental evaluation are discarded. More specifically, in order to assess the best value for such a threshold, we performed a sensitivity evaluation through a large-scale gradient descent algorithm [22] with learning rate $\zeta = 0.1$, thus finding a value of 0.3. The subjects of POIs are treated as tags with values of confidence set to 1.

To evaluate the probability that a user appreciates a given POI, this module takes advantage of her feedback formed by the Facebook pages tagged by the user with a *like*, as seen in Section 4.1.1. Since the user's *like* have been mapped to DBpedia entities using the previous procedure, the tag extraction method described above can be used for both POIs and Facebook pages. Therefore, we have one set of tags associated with the *liked* page and one associated with the POI to be evaluated. A metric for assessing the similarity and diversity of sample sets is the *Jaccard coefficient*. Given two sets A and B, it is defined as the ratio between the size of their intersection set and the size of their union set:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

The Jaccard coefficient can have a value between 0 and 1, where 0 indicates the total diversity and 1 the total correspondence between the two sets.

Vector-based selection (V) In this module, each POI is evaluated based on the similarity between its representation and the representations of the Facebook pages extracted from the user's *like*. The difference between the two modules lies in the way of representing the DBpedia entity: in this case we consider vectors instead of tags. To this aim, we take advantage of the *Word2Vec* [23] algorithm, which makes use of neural networks to represent entities in a vector space. The result is a multi-dimensional array able to map each dictionary term to a vector that composes the vector space. In a well-trained model, not only similar words are grouped close together as clusters (and, therefore, show high similarity values), but all words in a cluster are more or less equidistant with words having similar relationships.

[9] https://tagme.d4science.org/tagme/

The classic example is the following: the words *Rome*, *Paris*, and *Berlin* will be close together and each of them will have similar distance in the vector space from the countries of which they represent the capitals, that is, *Italy*, *France*, and *Germany*. This aspect allows us to make use of sums and subtractions of vectors to move from one word to another following a logical reasoning as the following one:

$$Rome - Italy + Germany = Berlin$$

For this application, the *Word2Vec* model was trained using the terms of the Wikipedia articles. In order to find the vector associated with a text, instead of a single word, we used *Doc2Vec* [24] (i.e., an extension of *Word2Vec* applied to documents instead of words). The vectors of words contained in the text are extracted and summed (alternatively, the arithmetic mean can be used). This solution is acceptable for short texts, as it is in our case, but not for longer texts in which the continuous sums would end up canceling each other, so leaving only noise. Through this strategy it is possible to represent a whole text by means of a single vector. In input to the *Word2Vec* algorithm, the `dbo:abstract` property associated with the DBpedia entities is used as the basis for the object representation. Having one vector associated with a user's *like* and one associated with the POI to be evaluated, it is possible to evaluate their similarity through the *cosine similarity*. Given two vectors $\overrightarrow{A}$ and $\overrightarrow{B}$, their cosine similarity can be calculated as follows:

$$\frac{\sum_{k=1}^{n} A(k)B(k)}{\sqrt{\sum_{k=1}^{n} A(k)^2}\ \sqrt{\sum_{k=1}^{n} B(k)^2}} \tag{4}$$

and is equal to the cosine of the angle formed by the two vectors. In this case, the result is a value between -1 and $+1$. A value of 0 denotes two vectors with different "meanings," while negative values denote components in relation, but with opposite "meaning."

Integration-based selection (I) The last module integrates the information coming from the analysis of the user's activity on social media with that related to her clicks on the images during the registration process. To this aim, the module maps the categories of Facebook *liked* pages in the ten categories used in the system to classify POIs (see Section 4). In this way, each user's Facebook *liked* page (for whom a correspondence with the POI category has been identified) contributes to her user profile in the same way as any user's click on the images shown to her in the registration form. We recall that the user profile is represented as a vector of ten components, one for each POI category, expressing the user's interest in it. Then, the user profile vector is compared with each of the POI vectors extracted by LOD for assessing its relevance to the target user. The calculation of similarity is performed through Eq. (2).

4.1.4 Presentation and evaluation of results

Once POIs deemed most relevant to the target user have been calculated, the system has a list of 60 POIs (ten for each recommendation strategy) to present to the user. For each POI, the user can view some information such as its name, description, image, position relative to the center of the search area, and distance in kilometers, and can, if necessary, be redirected to the associated Wikipedia page (for more details, see Section 4.3). The user is asked to examine the list and assign a rating to each element according to a 5-point Likert scale. One of the advantages of this scale is the ease with which it is possible examine data that, once expressed in numerical form, can also support the statistical analyses described in Section 5.3.

4.2 Data exploration

A module accessible from the results visualization page allows the user to explore LOD associated with the proposed point of interest. This is an accessory function that the user can exploit to increase her knowledge about a POI before evaluating it, and leads to the dynamic generation of a data oriented graph. The nodes correspond to DBpedia entities, the edges correspond to the predicates that link one entity to another. The architecture of this component has a part of user interaction with the interface that causes the invocation of asynchronous calls (AJAX) for LOD recovery from the Web, and a part of graph construction and update through the D3.js[10] library. The user acts by carrying out the following two operations:

1. Selection of one node of the graph to browse the list of predicates associated with it;
2. Expansion of one predicate of the list that leads to the introduction of a new node and edge in the graph.

Potentially, the user can continue expanding data, even if the information is less and less important as it moves away from the starting point. Furthermore, a self-expansion functionality is available, which navigates and expands nodes taking into account the user profile. It is essentially an application of the content-based recommendation strategy to the domain of data browsing. This module considers in turn the elements "reachable" by an entity and decides whether to introduce it or not in the graph based on its relevance to the user. The relevance is established not only based on the properties of the object to be introduced, but also on those of its immediate neighbors. The idea behind this algorithm is that an entity may not be immediately relevant to the user, but its expansion could lead to information that is instead.

[10] https://d3js.org/

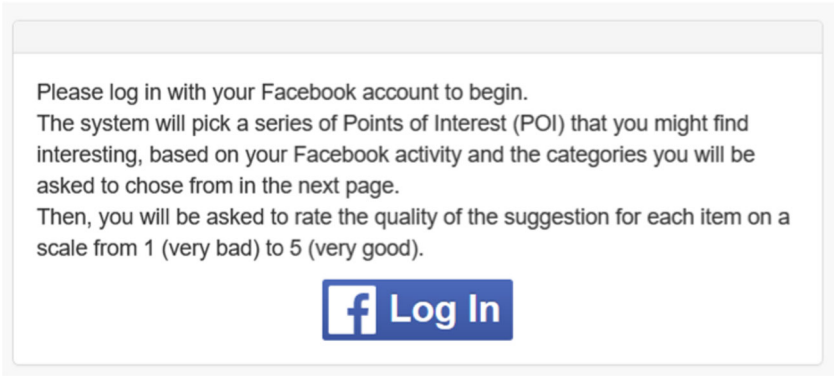

Fig. 4 Login via Facebook

The aim of the LOD exploration module is to provide the user with tools to explore the LOD graph. In this way, she can gather more elements to judge if that POI can actually be of interest to her. Furthermore, the module shows the potential of LOD for the purposes, for example, of the cross-domain recommendation.

4.3 User interface

This section describes the main components of the graphical interface and explains how the user can interact with the system.

In Fig. 4, the login panel via Facebook is shown, which reports instructions for the system evaluation. The login button opens the standard pop-up for accessing Facebook. Subsequently, the user can choose between a series of images clicking on those she considers interesting (see Fig. 5). Figure 6 depicts the page where the user can select the search area by placing a marker on the map. The user can also place the marker on the map by entering an address in the search bar located in the upper left corner. The system also provides the user with the standard panning, scaling, and zooming operations of the map, as well as auto-completion functionality for address search. The slider allows the user to adjust the value of the search radius r. Once the search is finished, the user is redirected to the page shown in Fig. 7. In the left panel, POIs are shown with a marker. The color of the marker denotes the category to which it belongs. In the right panel, the same results are shown in an accordion list and can be expanded to view

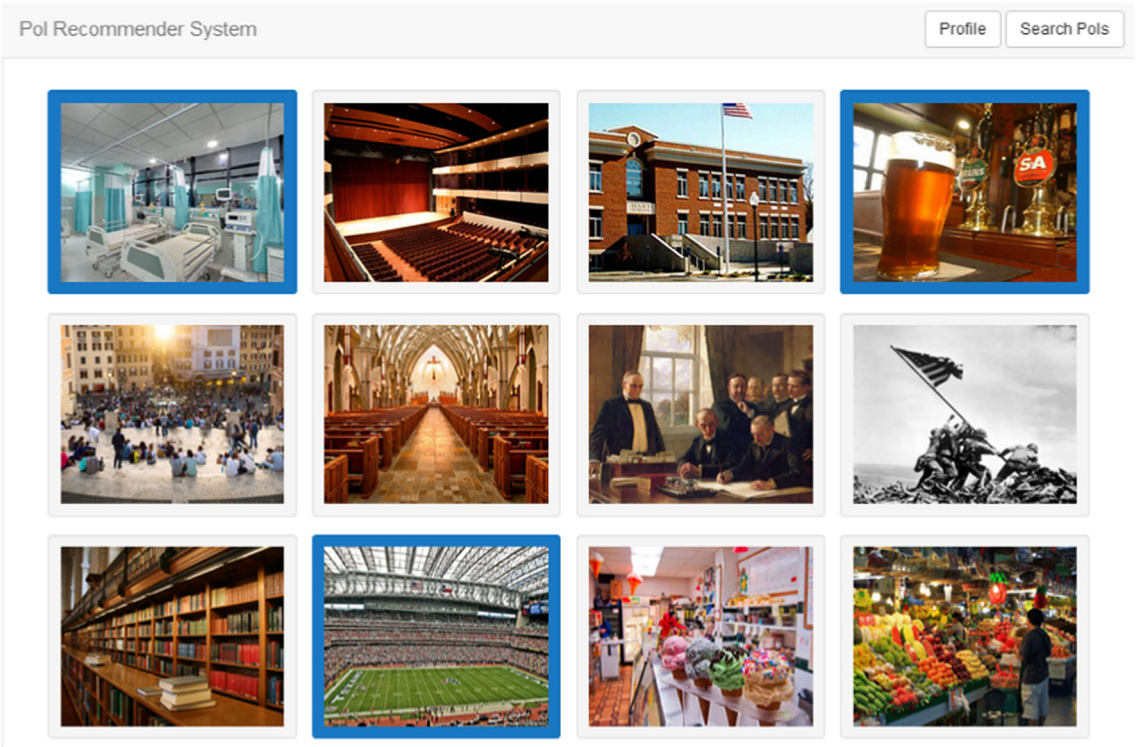

Fig. 5 Pictures form

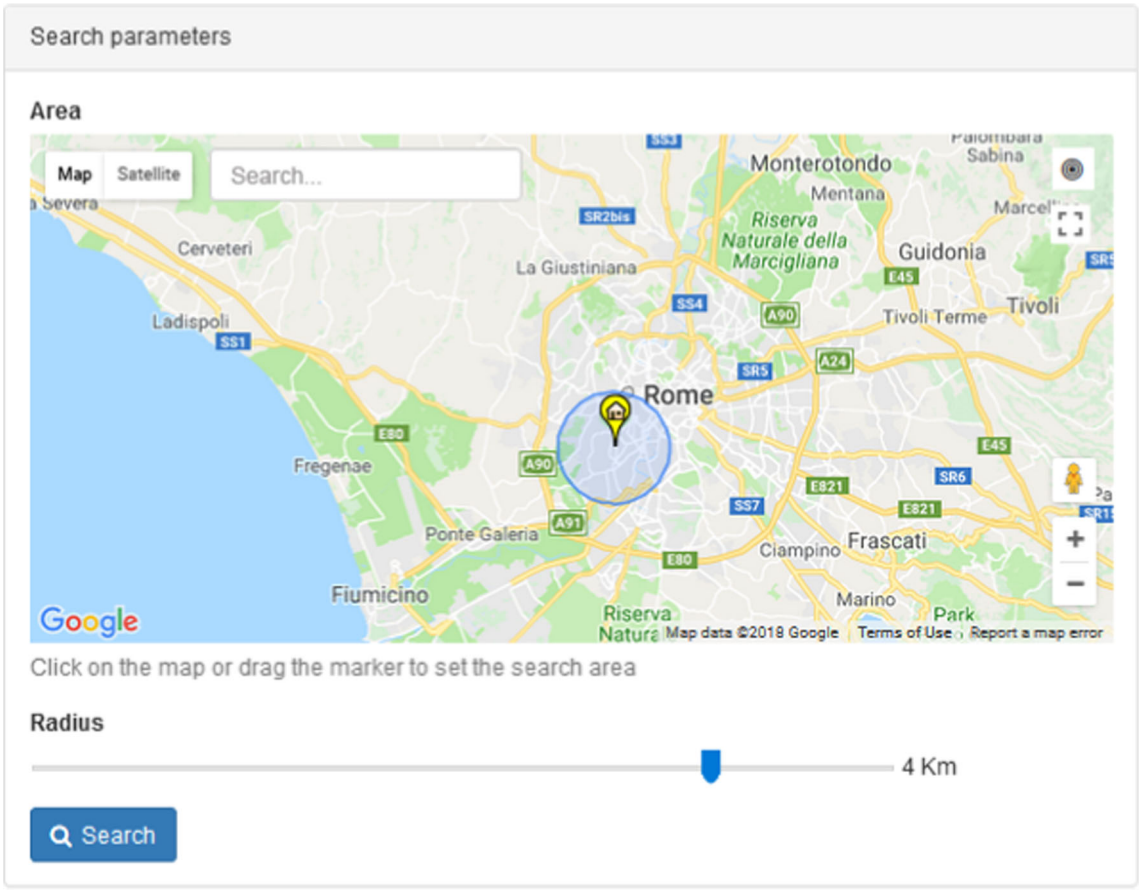

Fig. 6 Search parameters specification

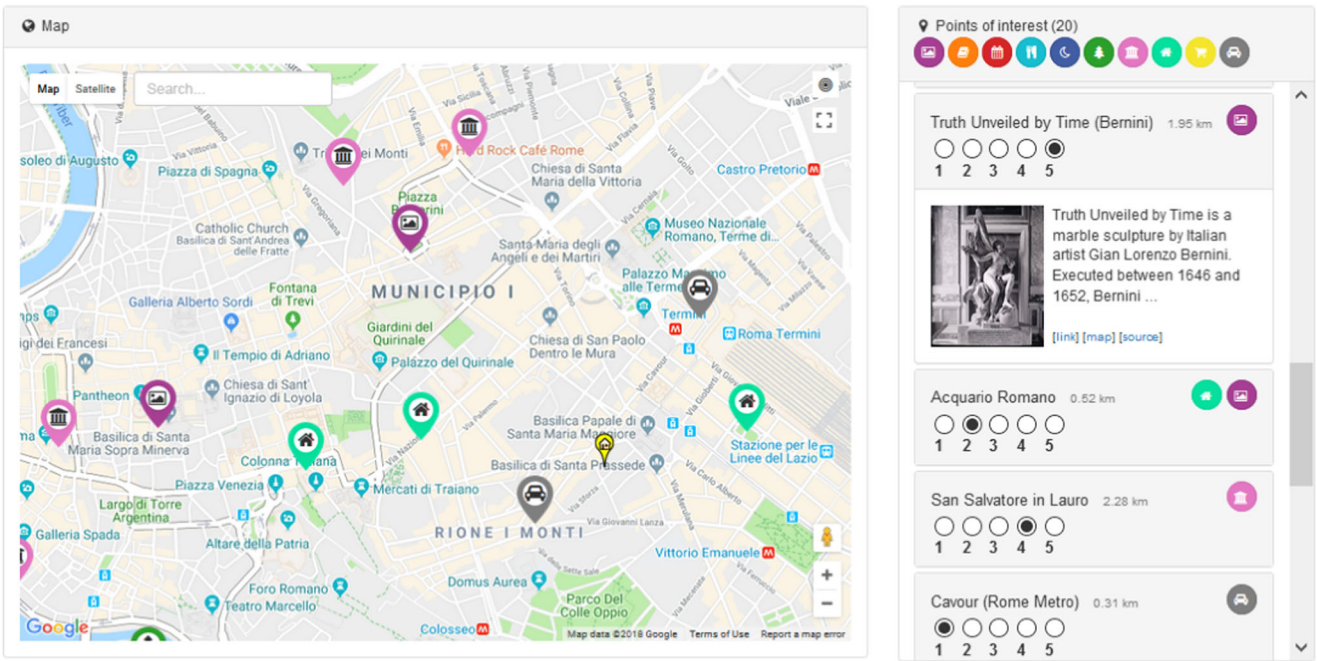

Fig. 7 Results visualization

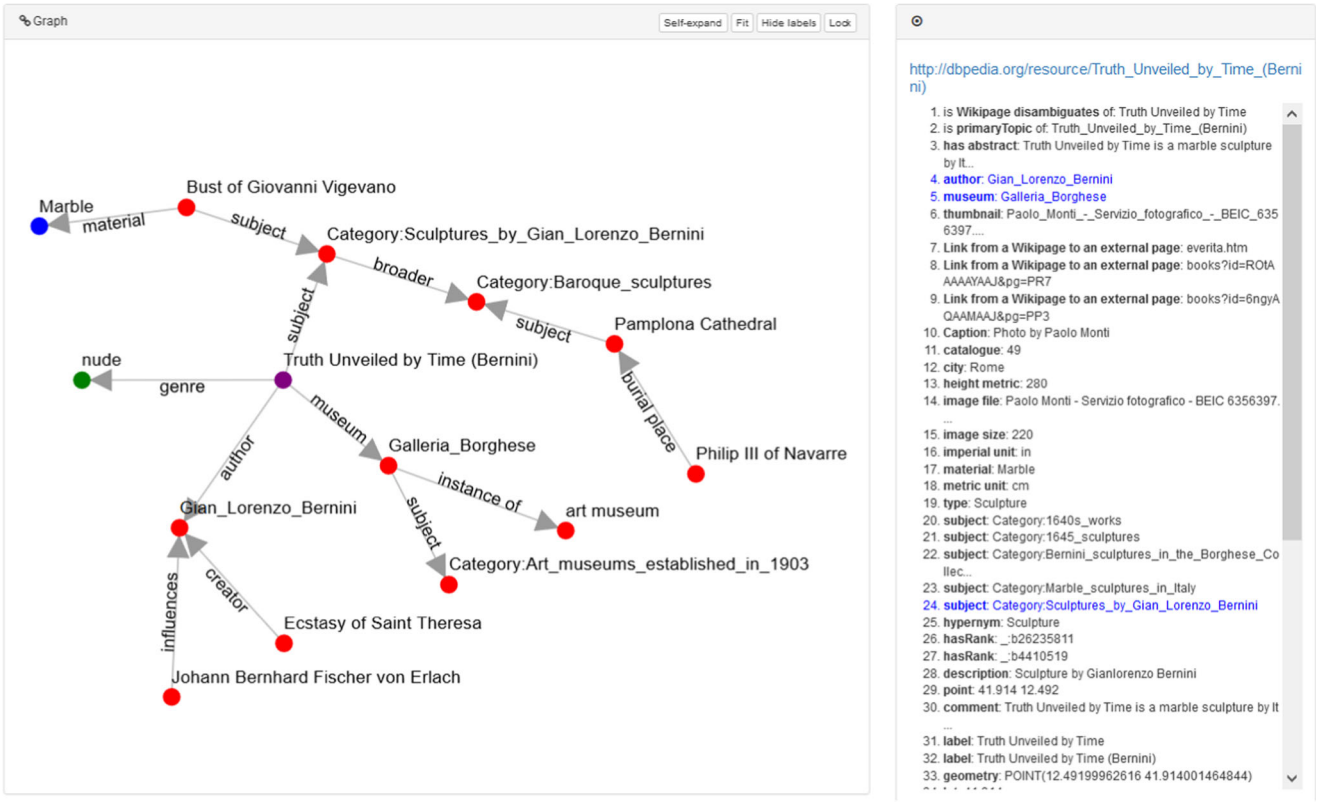

Fig. 8 LOD graph generation

the details available. From here, the user can also filter POIs by category and evaluate them in terms of the Likert scale. Finally, the LOD exploration module through graph generation is shown in Fig. 8. The colors of the nodes identify the starting element (violet), expandable nodes (red), fully expanded nodes (green), and literal nodes (blue). On the right side, the list of predicates of the currently selected node is presented, through which the user can add new elements to the graph (the predicates already added to the graph are shown in blue).

5 Experimental evaluation

This section reports the followed experimental strategy and the obtained findings.

5.1 User study

A user study was performed to evaluate the system performance. The participants were 75, all of them with a valid Facebook account and at least 30 *like*. The average number of *like* per user was 87.91. Their demographic characteristics are shown in Table 3. Testers were asked to evaluate a list of 60 POIs suggested according to the six strategies defined in Section 4.1.3. The user could choose both the location and the value of the r radius within which to search for the candidate POIs. In a first version of the system, the user could also choose the algorithm to be adopted to generate the list of suggested POIs. This solution raised two problems:

1. The user could be biased towards a certain algorithm and attribute non-objective evaluations to the quality of the proposed POIs;
2. The list produced by a single algorithm could be affected by a scarce variety of objects.

Table 3 Characteristics of testers

	Item	Frequency	Percentage
Gender	Female	34	45%
	Male	41	55%
Age	18–30	45	60%
	31–50	19	25%
	51–70	11	15%
Profession	Student	43	57%
	Teacher	14	19%
	Employee	10	13%
	Freelancer	5	7%
	Unemployed	3	4%

To address those issues, we decided to combine the outputs of the various modules into a single unordered list of items. In this way, the user could not be biased towards a particular system, being not aware of the list from which the individual POI comes from. For each of the six lists obtained through the various recommendations strategies described below, the ten most relevant POIs were extracted and combined into a single list of 60 elements arranged with no order.

The achieved results are shown in Fig. 9. It can be noticed that the integration-based strategy (*I*) allowed the system to obtain the best performance, followed by the content-based (*C*), the vector-based (*V*), and the tag-based (*T*). Then, the popularity-based (*P*) and random (*R*) strategies follow.

In the light of these experimental results, some considerations can be made. First of all, those findings show that taking into account both the user's activity on social media and her clicks on images gives better performance than considering only part of such an information. The strategy based on the *Word2Vec* algorithm enabled the system to achieve comparable results with those of the content-based strategy. Differently, the tag-based strategy was not so effective. Evidently, using vectors (obtained through *Word2Vec*) instead of tags (obtained through an annotation service) to represent the DBpedia entities allows for a better identification of the relationships between the user's *like* and the candidate POIs. Furthermore, it can be noted that all the strategies that consider the user profile offer better performance than non-personalized ones. Nevertheless, the popularity-based strategy that indiscriminately suggests the most popular POIs to all users obtained significant results, proving once again that users are often satisfied when the most popular items are suggested to them.

5.2 Discounted cumulative gain

The *discounted cumulative gain (DCG)* [25] is a measure of the quality of a ranking of objects. This metric is often used in Information Retrieval for the evaluation of search engines. It relies on two assumptions:

1. Highly relevant documents are more useful than marginally relevant documents;
2. If a relevant document receives a low ranking (and, therefore, is located further away from the beginning of the list), it is also less useful for the user because it is less likely to be examined.

The *DCG* at a particular p location of the document list is calculated through the following formula:

$$DCG@p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{5}$$

where rel_i returns the relevance at position i. The relevance of a document (numerator) is penalized proportionally to the logarithm of its position (denominator). However, the calculated value is not particularly expressive and cannot be used to compare two lists of different lengths. For this purpose, the *normalized DCG* or *NDCG* is introduced, which maps the calculated value in a range from 0 to 1. This metric is defined as follows:

$$NDCG@p = \frac{DCG@p}{IDCG@p} \tag{6}$$

where *IDCG@p* is the ideal *DCG* calculated on a ranking sorted by decreasing relevance, namely, the ranking that, hypothetically, would guarantee the best results.

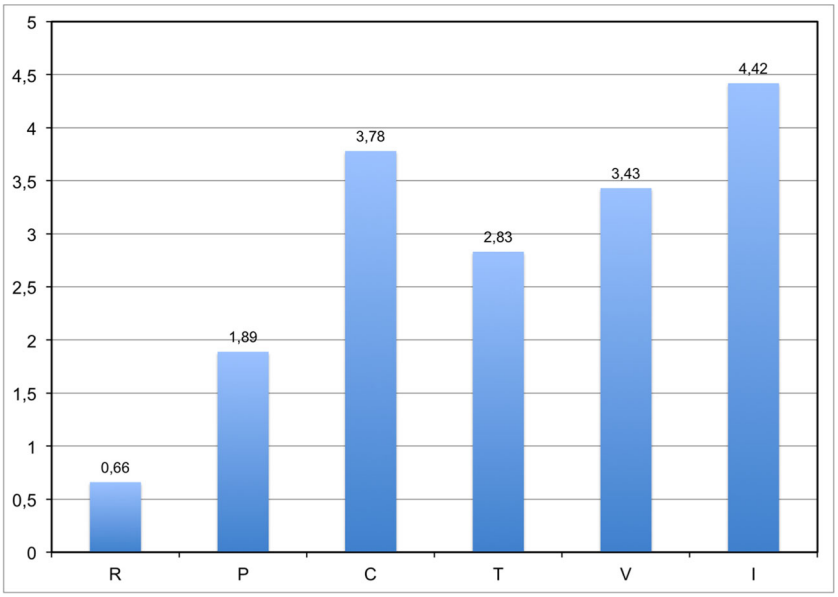

Fig. 9 Rating values using the six following recommendation algorithms: random (R), popularity-based (P), content-based (C), tag-based (T), vector-based (V), and integration-based (I)

For the system evaluation, we imagined to use a single strategy to recommend POIs at a time and we calculated the value of *NDCG* from the first to the tenth position of the lists obtained in this way. The trend of the various recommendation strategies is shown in Fig. 10. Consistently with the results described above, it can be noted how the content-based (*C*) and the integration-based (*I*) strategies had the best results, starting with values equal to 0.64 and 0.74 and constantly increasing up to 0.79 and 0.91, respectively. Next, the vector-based (*V*), the tag-based (*T*), the popularity-based (*P*), and the random (*R*) strategies. In particular, it can be observed that the popular-based strategy grows more sharply than previous ones, but with much lower values. Overall, there is a clear difference between the personalized recommendation strategies and the other two approaches.

5.3 Statistical significance test

The simple observation of the rating average and the values of $NDCG@p$ is not, however, sufficient to be able to declare with certainty that a recommendation strategy offers better performance than another. In fact, the obtained results could be due solely to chance and not be indicative. This is the so-called *null hypothesis* that has to be rejected so that the results can be considered significant also from the statistical point of view. For this purpose, the *t test* is introduced, so called because it is based on the *t-value* calculation: a statistic that allows the used sample of data to be summarized through a single numerical value. The computation compares the sample mean with the one of the null hypothesis by taking into account the variance and number of data. The formula for calculating the *t-value* is as follows:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{{s_1}^2}{N_1} + \frac{{s_2}^2}{N_2}}} \tag{7}$$

where $\overline{X_1}$, $\overline{X_2}$, ${s_1}^2$, ${s_2}^2$, N_1, and N_2 are the arithmetic mean, variance, and number, respectively, for the first and second sample. The probability distribution of the possible values of t is well known in the statistical literature and is, therefore, easily traceable if its *degrees of freedom* are known, a value closely related to the used data sample and can be computed as follows:

$$v = \frac{\left(\frac{{s_1}^2}{N_1} + \frac{{s_2}^2}{N_2}\right)^2}{\frac{{s_1}^4}{{N_1}^2(N_1-1)} + \frac{{s_2}^4}{{N_2}^2(N_2-1)}} \tag{8}$$

Having calculated the *t-value* and the form of the distribution of t based on the degrees of freedom, it is possible to know how likely it is to obtain a *t-value* from a sample of data, assuming that the null hypothesis is valid. That is, we calculate the integral (the area under the curve) for absolute values of t higher than the calculated *t-value*. The use of the absolute value is motivated by the fact that we intend to consider the difference of the sample from the null hypothesis in positive and negative direction. Should be noted how the distribution of t has its maximum in 0: this implies that as the value of t increases, the probability that the null hypothesis is valid decreases. For this kind of test, the validity of the null hypothesis is usually rejected for probability values lower than 0.01 (1%).

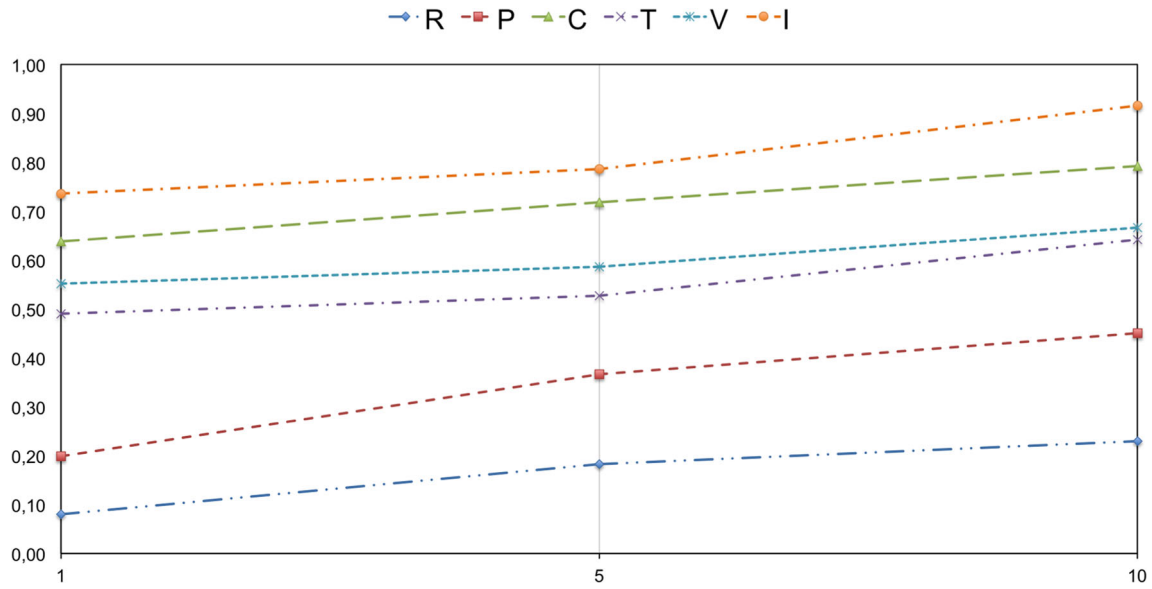

Fig. 10 $NDCG@p$ values with $p = 1, 5, 10$ for the six following recommendation algorithms: random (R), popularity-based (P), content-based (C), tag-based (T), vector-based (V), and integration-based (I)

For all the recommendation strategies, the probability value was much lower, so as to reject the null hypothesis.

6 Conclusions and future works

The objective of the research activities presented herein was to verify the possible benefits coming from the combination of social and linked open data for points of interest recommendation. To this aim, we have designed and realized a system able to provide the target user with personalized results based on the analysis of her Facebook profile, her clicks on a sample of selected images representing specific categories of points of interest, and the characteristics of the objects identified through LOD. The system is also able to obtain additional information for POIs that the user does not know through a semi-automated, guided exploration. The obtained experimental results allowed us to ascertain that the deployed recommendation model guarantees higher performance than recommenders in which the information related to the user profile is used only partially. The proposed strategies are, therefore, able to guide the active user during the selection phase in front of a large number of POIs available.

Among the possible future developments of the proposed recommender there is the implementation of other suggestion strategies, whose synergistic combination can lead to benefits higher than those obtained through this version of the system. More specifically, we plan to take advantage of collaborative strategies that have not been used in the proposed system. Furthermore, we would like to further exploit the potential of LOD for cross-domain and itinerary recommendation. Regarding the first point, our recommender could provide the target user with multimedia and textual content related to the POIs suggested to her by following the LOD semantic links. Concerning the second point, we would like to supply the active user with also context-aware personalized itineraries among POIs, in order to improve her experience [26, 27]. Moreover, we intend to develop new techniques for extracting further information from the analysis of the activity performed by the user and those belonging to her social graph, in addition to the pages tagged by her with a *like*. Finally, we plan to enrich the user profile with additional information about her personality [28, 29], as well as the temporal dynamics [30, 31] and the actual nature [32, 33] of her interests.

References

1. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York
2. Biancalana C, Gasparetti F, Micarelli A, Sansonetti G (2013) Social semantic query expansion. ACM Trans Intell Syst Technol 4(4):60:1–60:43
3. Ricci F, Rokach L, Shapira B (2015) Recommender systems handbook, 2nd edn. Springer Publishing Company Incorporated
4. Biancalana C, Gasparetti F, Micarelli A, Sansonetti G (2013) An approach to social recommendation for context-aware mobile services. ACM Trans Intell Syst Technol 4(1):10:1–10:31
5. Heitmann B, Hayes C (2010) Using linked data to build open, collaborative recommender systems. In: Linked Data Meets Artificial Intelligence, Papers from the 2010 AAAI Spring symposium, Technical Report SS-10-07, Stanford, California, USA, March 22–24, 2010. AAAI
6. Di Noia T, Ostuni VC (2015) Recommender systems and linked open data. In: Faber W, Paschke A (eds) Reasoning Web. Web logic rules: 11th International Summer School 2015, Berlin, Germany, July 31–August 4, 2015, Tutorial Lectures. Springer International Publishing, Cham, pp 88–113
7. Gasparetti F (2017) Personalization and context-awareness in social local search: state-of-the-art and future research challenges. Pervasive Mob Comput 38:446–473
8. Zhao S, King I, Lyu MR (2016) A survey of point-of-interest recommendation in location-based social networks. CoRR
9. Yang D, Zhang D, Yu Z, Wang Z (2013) A sentiment-enhanced personalized location recommendation system. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media. HT '13. ACM, New York, pp 119–128
10. Gurini DF, Gasparetti F, Micarelli A, Sansonetti G (2018) Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization. Futur Gener Comput Syst 78:430–439
11. Yang D, Zhang D, Yu Z, Yu Z (2013) Fine-grained preference-aware location search leveraging crowdsourced digital footprints from LBSNs. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. UbiComp '13. ACM, New York, pp 479–488
12. Yang D, Zhang D, Zheng VW, Yu Z (2015) Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Trans Syst Man Cybern: Syst 45(1):129–142
13. Sansonetti G, Gurini DF, Gasparetti F, Micarelli A (2017) Dynamic social recommendation. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. ASONAM '17. ACM, New York, pp 943–947
14. Bizer C, Heath T, Berners-Lee T (2009) Linked data—the story so far. Int J Semantic Web Inf Syst 5(3):1–22
15. Passant A (2010) Measuring semantic distance on linking data and using it for resources recommendations. In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, Palo Alto. AAAI Press, California, pp 93–98
16. Wang Y, Stash N, Aroyo L, Hollink L, Schreiber G (2009) Semantic relations for content-based recommendations. In: Proceedings of the 5th International Conference on Knowledge Capture. K-CAP '09. ACM, New York, pp 209–210
17. Di Noia T, Mirizzi R, Ostuni VC, Romito D, Zanker M (2012) Linked open data to support content-based recommender systems. In: Proceedings of the 8th International Conference on Semantic Systems. I-SEMANTICS '12. ACM, New York, pp 1–8
18. Lo Bue A, Wecker AJ, Kuflik T, Machì A, Stock O (2015) Providing personalized cultural heritage information for the smart region—a proposed methodology. In: Cristea AI, Masthoff J, Said A, Tintarev N (eds) Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015),

Dublin, Ireland, June 29–July 3, 2015. Volume 1388 of CEUR Workshop Proceedings, CEUR-WS.org, pp 1–7
19. Cantador I, Bellogin A, Castells P (2008) A multilayer ontology-based hybrid recommendation model. AI Commun 21(2–3):203–210
20. De Angelis A, Gasparetti F, Micarelli A, Sansonetti G (2017) A social cultural recommender based on linked open data. ACM, New York, pp 329–332
21. Sansonetti G, Gasparetti F, Micarelli A, Cena F, Gena C (2019) Enhancing cultural recommendations through social and linked open data. User Modeling and User-Adapted Interaction
22. Zhang T (2004) Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the 21st International Conference on Machine Learning. ACM, p 116
23. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13. Curran Associates Inc, pp 3111–3119
24. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. CoRR
25. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 20(4):422–446
26. Fogli A, Micarelli A, Sansonetti G (2018) Enhancing itinerary recommendation with linked open data. In: Stephanidis C (ed) HCI International 2018 – Posters' Extended Abstracts. Springer International Publishing, Cham, pp 32–39
27. Fogli A, Sansonetti G (2019) Exploiting semantics for context-aware itinerary recommendation. Personal and Ubiquitous Computing
28. Bologna C, De Rosa AC, De Vivo A, Gaeta M, Sansonetti G, Viserta V (2013) Personality-based recommendation in e-commerce. In: CEUR Workshop Proceedings. Volume 997 of CEUR Workshop Proceedings, Aachen, CEUR-WS.org
29. Onori M, Micarelli A, Sansonetti G (2016) A comparative analysis of personality-based music recommender systems. In: CEUR Workshop Proceedings. Volume 1680 of CEUR Workshop Proceedings, Aachen, CEUR-WS.org, pp 55–59
30. Arru G, Feltoni Gurini D, Gasparetti F, Micarelli A, Sansonetti G (2013) Signal-based user recommendation on Twitter. In: Proceedings of the 22nd International Conference on World Wide Web. WWW '13 Companion. ACM, New York, pp 941–944
31. Caldarelli S, Gurini DF, Micarelli A, Sansonetti G (2016) A signal-based approach to news recommendation. In: CEUR Workshop Proceedings. Volume 1618 of CEUR Workshop Proceedings, Aachen, CEUR-WS.org
32. Gurini DF, Gasparetti F, Micarelli A, Sansonetti G (2013) A sentiment-based approach to Twitter user recommendation. In: CEUR Workshop Proceedings. Volume 1066 of CEUR Workshop Proceedings, Aachen, Germany, CEUR-WS.org
33. Gurini DF, Gasparetti F, Micarelli A, Sansonetti G (2014) iSCUR: interest and sentiment-based community detection for user recommendation on Twitter. In: Dimitrova V, Kuflik T, Chin D, Ricci F, Dolog P, Houben GJ (eds) User modeling, adaptation, and personalization. Springer International Publishing, Cham, pp 314–319