Personal and Ubiquitous Computing (2019) 23:181–197
https://doi.org/10.1007/s00779-018-01194-w

ORIGINAL ARTICLE

LOOKER: a mobile, personalized recommender system in the tourism domain based on social media user-generated content

Sondess Missaoui[1,2] · Faten Kassem[3] · Marco Viviani[1] · Alessandra Agostini[1] · Rim Faiz[4] · Gabriella Pasi[1]

Received: 25 June 2018 / Accepted: 12 December 2018 / Published online: 4 January 2019

Abstract

In a ubiquitous computing scenario, characterized by pervasive technologies, tourists can get assistance from mobile technologies in planning their trips. In a context where more and more people own smartphones, tourists expect to get personalized suggestions just in time whenever and wherever they need. To be effective, mobile applications for travel recommendation should consider both the variability of the user's interests and an effective way to express them while interacting with the environment. This paper presents LOOKER, a mobile recommender system for tourism and travel-related services that considers the above-described issues. It is an adaptable application developed for the Android platform, which takes into account basic contextual information such as location and time, and implements a content-based filtering (CBF) strategy to make personalized suggestions based on the user's tourism-related user-generated content (UGC) s/he diffuses on social media. Specifically, the CBF strategy implemented in LOOKER is based on a multi-layer user profile, where the layers representing distinct travel-related service categories (e.g., restaurants, hotels, points of interest) are modeled via language models that are defined on the basis of the captured UGC. This allows inferring the interests and the opinions of travelers about the available items. To evaluate the usefulness and the usability of the LOOKER mobile application, user studies have been conducted. The positive outcomes that have been obtained illustrate the potentials of LOOKER.

Keywords Mobile recommender systems · Content-based filtering · Personalization · Language models · User-generated content · Social media

1 Introduction

Mobile technologies have recently evolved to significantly influence the user experience. Smartphones, in particular, offer an environment for a multitude of social media applications, especially in the tourism domain [61]. According to Kenteris et al. [36], "the concept of 'mobile tourism' has recently emerged wherein users access tourist content through mobile devices". In this scenario, a considerable amount of research is focusing on the development of *travel recommender systems* [26] to provide tourists with suitable travel-related recommendations. These systems are based on different models and follow different approaches; in recent years, a lot of attention has been paid, in particular, to the design of effective and user-friendly mobile applications for the tourism industry, to address a wide variety of tourists' information needs [17].

Within this area, the importance of contextual information, which can be easily captured via mobile technologies, has been recognized by researchers as the dominant factor affecting the user's decisions. For this reason, a number of recent studies have proposed *Context-Aware Recommender Systems* (CARS) [3], which provide recommendations that are adapted to the user's changing context. Context can refer to simple spatio-temporal information, such as the user's location and time, or to more detailed information, such as the entities surrounding the user (e.g, devices, services, and persons), or particular environmental characteristics (e.g., light, humidity, noise) [25]. The wide spectrum of CARS that have been proposed over the last decade in the travel recommendation scenario offers a context modeling that encompasses *location-aware* [33, 40], *time-aware* [75], and *multi-dimensional* CARS [14, 56].

✉ Marco Viviani
marco.viviani@disco.unimib.it

Extended author information available on the last page of the article.

In addition to contextual information, mobile apps make it possible to take advantage of the growing amount of *user-generated content* (UGC) that is diffused through social media by users. UGC, for example in the form of opinionated *on-line reviews*, can represent an incredibly rich source of information that can be employed to provide both decision support [70] and recommendations to on-line tourists [44].

By considering the above-described aspects, this work proposes LOOKER, a mobile recommender system developed for the Tunisian market. LOOKER provides personalized suggestions of *travel-related services*, namely, TR services, by taking into account both basic contextual information and the user's interests extracted from her/his travel-related user-generated content. Specifically, the system implements a *content-based filtering* (CBF) strategy that employs a user profile representation based on *language models* [53]; a user profile is built by using the content that the user has generated on social media, i.e., the reviews previously provided with respect to TR services that s/he liked. This UGC represents an explicit evidence about the user's preferences that have been specifically collected on Facebook,[1] Twitter,[2] and TripAdvisor.[3] These social media are widely used in Tunisia to share opinions in general, and on TR services in particular, usually in the form of on-line reviews. LOOKER has been fully developed for the Android operating system; two user studies have been conducted to evaluate both the usefulness and the usability of the mobile application, as well as the effectiveness of the provided recommendations.

The rest of the paper is organized as follows. Section 2 provides an overview of the related works concerning recommender systems with particular reference to the tourism domain, and the role of smartphones and mobile applications in mediating the tourist's experience. Section 3 illustrates in detail the theoretical basis of the proposed travel recommender system. The implementation of the mobile application and the preliminary evaluations are detailed respectively in Section 4 and Section 5. Finally, Section 6 concludes the paper and discusses some open issues and some future research directions.

2 Related works

To deal with the *information overload* problem, increased by the rapid spread of mobile devices, ubiquitous computing, and social media, there has been an increasing popularity of *Recommender Systems* (RS) [52], many of which have been developed in the tourism domain [62].

Recommender systems are a popular category of systems that provide a personalized suggestion of items to users based on their preferences [10]. RS basically rely on an *Information Filtering* (IF) algorithm. Two majors categories of IF exist: *collaborative filtering* (CF) and *content-based filtering* (CBF). In CF approaches, the recommendations are generated based on the prediction of items that a user is likely to prefer based on the items that people with similar preferences would prefer (*user-based* CF) or based on the similarity between the user's previously rated items and new items to be recommended (*item-based* CF). In CBF approaches, the user's preferences are formally represented in a *user model*, also called *user profile*. A CBF system selects items based on the similarity between the content of the items and the user profile, to recommend to the user those items that may satisfy her/his expectations. The important difference between CF and CBF algorithms is the type of information used to produce suitable recommendations for a user. While CF predicts the relevance of items by taking into account the user's behaviors and ratings, CBF is concerned with the content of items and user profiles. Despite the advances achieved by both CF and CBF in recommender systems, the recommendation accuracy may be affected by some important drawbacks. It is widely known that CF approaches may incur in the *cold start* problem [10], when the system does not have enough ratings to compute personalized recommendations [2]. Moreover, CBF could suffer from the *limited content analysis* problem [43], in the absence of sufficient meta-data and content associated with users and items.

Nowadays, advances in Web 2.0 technologies have promoted information sharing and collaboration among users. In this context, due to their popularity and real-time nature, social media can be considered as effective sources of information to address the above-described issues. The growing availability of user-generated content and associated meta-data (e.g., ratings, tags, friends, followers) has suggested to consider social information as a source of evidence for information filtering and seeking [45]. At the same time, mobile telecommunications technologies have seen the functionality of smartphones evolve to enable the detection of the user's context in different situations in a real-time manner [25].

In the following, the main approaches considering these aspects in the promising application domain of tourism will be illustrated, and their pros and cons will be discussed.

2.1 Recommender systems in the tourism domain

Several recommender systems in the tourism domain, usually known as *tourism recommender systems* or *travel recommender systems*, have been proposed over the years [29, 42, 44], with the aim of helping tourists in defining

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

[2] https://dev.twitter.com/overview/api/

[3] https://developer-tripadvisor.com/content-api/

their travel plans [19], and/or assisting them during their trips. In this context, an RS can be an intermediary between a tourist and a travel agency [42], or tourist guides that aim to facilitate the tourist decision-making process. Several content-based recommendation approaches focus on the matching between personal preferences against all the available travel-related services [11]. Early research efforts in this domain [42] required an explicit interaction of the user with the system, by providing her/his preferences, needs, characteristics, and even by exchanging textual messages [42]. In [67], for example, the *CityTrip* planner was proposed to visit five cities in Belgium. This system is essentially a tourist guide that provides a brief questionnaire to users, trying to obtain, in this way, their preferences and constraints. Nowadays, many popular social media or e-tourism Web sites develop their own RS for hotels, restaurants, museum or other *Points of Interest* (POI) recommendation, such as TripAdvisor. These on-line platforms also contain a social component allowing users to review and rate the system suggestions. Some popular travel Web agencies such as Expedia[4] and Trippy[5] use social components (e.g., the average ratings of other users) as informative data for collaborative recommendation to support tourists in their trip planning [2]. Nevertheless, traditional recommendation purposes in the tourism domain need to be improved, because during a trip the user will be confronted with a large amount of possible combinations of locations, events and activities. Moreover, when considering a new destination, the tourist will be confronted to specific constraints that should be taken into account to provide accurate recommendation, e.g., climatic situation, food constraints (vegetarian, halal, . . .), etc.

2.2 Context-aware recommender systems in the tourism domain

Recently, several works are dealing with a more challenging problem in the tourism domain, such as the user mobility and the effects of contextual information [68]. The advances of wireless communications and the rapid development of ubiquitous computing technologies create many new opportunities for RS. In fact, smartphones, through mobile applications, can enhance the tourist experience and offer the right services at the right time [66, 71]. In this context, *mobile recommender systems* [12, 18, 21] represent an important thread of research applied to tourism, since they can take advantage from several user's interactions with smartphones and social media applications.

This is a challenging task due to: (i) the variety of tourist's needs, and (ii) the competitiveness of the tourism industry [23], where developers have to come up with the most innovative and effective mobile app designed to make the touristic experience as easy, cheap, and fun as possible [35]. Therefore, the *app* market has recently seen the development of a wide range of mobile recommender systems [13, 29, 56, 65]. Many of these applications have improved their competitiveness by considering *context-awareness* in their filtering algorithms [3, 29]. Context-awareness [32], a core feature of ubiquitous and pervasive computing systems, has significantly evolved in RS to deliver accurate recommendations according to the user's current context [3]. The concept of context was initially intended as the geographic location only, which made *Location-Aware Recommender Systems* (LARS) the more widespread context-aware RS [40]. LARS privilege those services that are geographically closer to the target users, and ignore other several contextual information (e.g., weather, time, surrounding people, etc.) that can be modeled and utilized in the relevance assessment process. For instance, in [63] Tsai and Chung propose a location-aware RS where the location was employed as the main contextual information to develop a route recommendation system that supplies tourists with the POIs they should visit and in what order. In [7], another LARS, i.e., *Turist@*, is presented, which provides users with attractions based on their location. In [65], Tumas and Ricci describe *PECITAS*, a LARS for personalized point-to-point paths in the city of Bolzano, Italy. In addition to the geographic location, the system considers the user's travel-related preferences to recommend paths that pass through several attractions.

Nowadays, due to the increasing advances in sensor technology, the location does not represent anymore the main contextual information considered by RS. Hence, the context can be seen as any piece of interesting information regarding the user's current situation (physical, personal, or social) that can be gathered using sensing devices, allowing the system to automatically adapt according to the user's condition [25]. Over the years, many approaches for context modeling have been proposed (see for example [9]). In the tourist domain, *Context-Aware Recommender Systems* (CARS) can deal with any kinds of contextual information (i.e., location, time, weather, social and demographic context) [26, 50]. For example, the tourism recommender system *I'm feeling LoCo* [58] merges contextual information inferred from a user's social network profile and her/his mobile phone's sensors for place discovery. In [75], the authors exploit the "check-in" data to provide a time-aware POI recommendation. They recommend POIs to a given user at a specific time in a day. Similarly, in [59] Sebastia et al. propose a travel RS that provides the tourist with a list of the places that are likely to be of interest to her/him.

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

[5] http://www.trippy.com/

This RS employs the user's demographic information, time, activities, likes, and preferences from previous trips, as well as from the current trip.

In [3], Adomavicius and Tuzhilin have defined three algorithmic paradigms for incorporating contextual information into the recommendation process. Thus, approaches to develop CARS have been classified as: (i) *contextual pre-filtering* approaches, which use contextual information to filter out irrelevant items and then use information filtering algorithms to generate recommendations; (ii) *contextual post-filtering* approaches, which generate recommendations and then re-rank the suggestions according to the considered contextual information; (iii) *contextual modeling* approaches, which make use of contextual information directly within the recommendation process to generate suggestions.[6]

Table 1 summarizes CARS approaches, including the approach proposed in this paper, with respect to the IF algorithm implemented, the paradigm employed for incorporating context, and the contextual dimensions considered. With respect to state-of-the-art solutions, the proposed approach focuses on the aspect of considering how the user's opinions are expressed in social media with respect to a variety of travel-related services, such as restaurants, accommodations, flights, and cultural point of interest. In particular, the approach considers the way in which users employ the language in social media posts. The recommendation model at the bases of LOOKER is detailed in the next section.

3 LOOKER: description of the recommendation model

As it emerges from Table 1, the recommendation model of LOOKER is based on the contextual pre-filtering paradigm, where contextual information is used to filter out irrelevant items before the application of the CBF algorithm that considers the user's preferences.[7] For this reason, two distinct modules are at the basis of the proposed model: (i) a *spatio-temporal filtering module* and (ii) a *content-based filtering module*. The first module selects, among the travel-related services available, those which are suitable to the target user based on her/his location and on the TR service opening hours. The second module applies a content-based filtering algorithm that employs the user's preferences extracted from travel-related UGC to build a user profile. Based on these two modules, LOOKER is able to identify the nearest open travel-related services with respect to the user's geolocation. Then, it recommends the most relevant ones by taking into account the user's preferences. In LOOKER, four distinct categories of TR services are considered, namely "TR-service categories":

- *Food*, which includes restaurants and bars, coffee bars, and food trucks;
- *Shopping*, which includes fashion stores, bookstores, cosmetics and beauty supply, and children clothing;
- *Health*, which is related to healthcare services such as dentistry, nursing, medicine, optometry, midwifery, emergency, and hospitals;
- *Attractions*, which refers to points of interest for tourists such as beaches, national parks, mountains and forests, or cultural attractions including historical places, monuments, museums, and art galleries.

Details concerning the two modules are presented in the following Sections 3.1 and 3.2.

3.1 Spatio-temporal filtering module

In the spatio-temporal filtering module, the subset of the closest TR services to the target user, which are open at the current time (i.e., intended as the current day and a 2-h time-window starting from the current time), are considered for recommendation. This aims at selecting only relevant items (with respect to the spatio-temporal context) to be further analyzed with respect to the user's interests.

In this module, the distance between the user's location and the TR service is simply computed using the geographic coordinates (i.e., GPS coordinates, latitude, and longitude) captured by the user's smartphone (with the user's consent). A subset of TR services is selected in a predefined radius (set a priori by the user) around the user's geographic position using the *Google Places* API[8] (the user's position is updated every time the user interacts with the application). Furthermore, also opening hours of TR services are used as a filter, to restrict the subset of selected items and to avoid the unnecessary recommendation of TR services that are not effectively accessible to the user.

Within the obtained subset of TR services, the content-based filtering module described below selects those services that are more similar to the target user's interests.

3.2 Content-based filtering module

The content-based filtering module exploits the content generated by users in the tourism context as a source of information for representing users' preferences and building

[6] Recent empirical analyses by Panniello et al. [51] demonstrated that no CARS paradigm in particular dominates the others across all domains.

[7] Hence, diminishing the number of similarity comparisons that should be made between the user profile and the possible items to be recommended.

[8] https://developers.google.com/places/

Table 1 Comparison of the proposed model against related work approaches

Work	Items and Services recommended	Method		CA Approach			Contextual Information
		CBF	CF	Pre	Post	CM	
POST-VIA 360 [21]	POIs	✓				✓	User location, opinions, visited POIs
Mobile CARS [18]	Shopping	✓				✓	User location, social media data
POI group recommendation [80]	POIs			✓		✓	User location, season, time, weather, companionship, transport
CRUMPET [54]	Travel tips, tour suggestions					✓	User location, visited POIs
MTRS [28]	POIs		✓				User location, time, weather, user's mobility history, peer user's ratings
I'm feeling LoCo [58]	Restaurants, hotels, bars, walking trails	✓					User location, transport, user's mood
MobyRek [57]	Restaurants	✓	✓		✓	✓	User location, opinions
iTravel [73]	On-tour attractions		✓	✓			User location
Turist@ [7]	Tourist attractions	✓	✓	✓			User location
PlanTour [19]	Tourist plans, restaurants, POIs		✓			✓	User location, time
eTourism [59]	Tourist plans, list of activities	✓			✓		User location, time, user activity information (e.g., duration of the activity)
Proposed Approach	Food, shopping, health, attractions	✓		✓			User location, time

Context-aware approaches: Pre: pre-filtering; Post: post-filtering; CM: contextual modeling

Filtering algorithm: CBF: content-based filtering; CF: collaborative filtering

user profiles. Specifically, to build each user profile, this module takes into account the *textual content* in the form of *on-line reviews* that each target user has previously provided with respect to distinct TR services on different social media.

Here below, the main components of the content-based filtering module are described: (i) the *multi-layer user profile*, (ii) the *TR-service profile*, and (iii) the *content-based filtering algorithm* which is used for the final recommendation of TR services.

3.2.1 Multi-layer user profile

The CBF module employs a *multi-layer user profile*, based on the idea that each layer represents the user's interests with respect to a distinct TR-service category. As detailed in Section 3, four TR-service categories are considered by the proposed mobile application: *food*, *shopping*, *health*, and *attractions*. *Statistical language modeling* [53] has been employed in each layer, to profile a user based on the on-line reviews s/he has written about TR services in a given category, and collected from the user's social media accounts (with her/his consent). The idea of employing the content written by a user and language models to build user profiles has been taken from the literature [74]; in this paper, adopting a multi-layer representation of the user profile based on language models allows to exploit the different vocabulary employed by the user to review TR services in different TR-service categories, hence capturing different interests per category.

In LOOKER, by following a *positive filtering* strategy, only "positive" reviews reflecting the user's positive feedbacks are employed to build the user profile; the rationale behind this choice is to directly identify and recommend only those TR services that are more similar to what the user likes.[9] To detect a positive review, it is possible to use either the *ratings* associated with the review, i.e., ratings in the form of "stars" (in a [1–5] range) that provide an evaluation of the considered TR service, or the content of the review. When ratings are provided, a review is considered as positive if the assigned evaluation is greater than or equal to 3 stars. In the absence of ratings, a simple *polarity analysis* that takes into account the ratio between "positive" and "negative" terms that appear in the review is performed.[10]

Formally, for a given target user u, a set $R_c = \bigcup r_i$ of the user's positive reviews about a given TR-service category c is identified. The *user profile*, denoted as θ_u, is composed of many distinct *layers* – one for each considered TR-service category – each of which is built by generating a *language model* based on the user's reviews belonging to R_c. In particular, statistical language models based on *unigrams* are generated [53]. Each layer of the user profile is formally denoted as θ_c, and is estimated, by taking inspiration from [74], as follows:

$$\theta_c = P(w|R_c) = \frac{1}{\mid R_c \mid} \textstyle\sum_{r_i \in R_c} P(w|r_i) \qquad (1)$$

In Equation (1), w represents a given word in the subset of reviews R_c, and $P(w|r_i)$ is estimated by using a language model for r_i via *Dirichlet prior smoothing* [76] as follows:

$$P(w|r_i) = \frac{nocc(w,r_i) + \mu \frac{nocc(w,R_c)}{\sum_w nocc(w,R_c)}}{\sum_w nocc(w,r_i) + \mu} \qquad (2)$$

where $nocc(w, r_i)$ is the number of occurrences of the word w in the review r_i, μ is a smoothing parameter, and

[9] This strategy could be coupled in future works with *negative filtering*, i.e., the process of filtering out those TR services that the user dislikes.

[10] https://github.com/Ruthwik/Sentiment-Analysis/

$nocc(w, R_c)$ represents the number of occurrences of the word w in R_c.

3.2.2 TR-service profile

To evaluate the interest of a user about a target TR service, the user profile must be compared with the profile of the service. Specifically, the task of evaluating the similarity between the profile of a user u and the profile of a TR service s belonging to the TR-service category c, becomes the task of evaluating the similarity between the user profile layer θ_c with the *TR-service profile*, denoted by θ_s. In the proposed approach, θ_s is based on the positive reviews provided by other users on the considered service.

Formally, given $R_s = \bigcup r_j$ the set of positive reviews written about a TR service s, and r_j a specific review written about the TR service, the TR-service profile is estimated according to (1), where r_j replaces r_i, and R_s replaces R_c. The probability of generating a word w given a review r_j is calculated as in (2), where r_j replaces r_i, and R_s replaces R_c.

3.2.3 Content-based filtering algorithm

The content-based filtering algorithm, by using the previously defined user and TR-service profiles, estimates a *relevance score* for each of the TR services considered, by comparing the user profile (i.e., the layer corresponding to the category c) with the TR-service profile. In fact, based on the way in which both user and TR-service profiles have been defined, the problem of recommending items to users becomes the task of evaluating the *similarity* between the two profiles, in particular, the distribution θ_c of the user profile, and the distribution θ_s related to the travel-related service s belonging to the same category c. To compute the relevance score $\hat{r}_{u,s,c}$, as proposed in [47], the *Kullback-Leibler* (KL) divergence [8, 39] has been employed as follows:

$$\hat{r}_{u,s,c} = \frac{1}{D_{KL}\left(\theta_c \| \theta_s\right)} \tag{3}$$

where $D_{KL}\left(\theta_c \| \theta_s\right)$ corresponds to the divergence between two probability distributions, which can be computed as follows:

$$D_{KL}\left(\theta_c \| \theta_s\right) = \sum_{w} P\left(w \mid R_s\right) log \frac{P\left(w \mid R_s\right)}{P\left(w \mid R_c\right)} \tag{4}$$

It is assumed that $D_{KL}\left(\theta_c \| \theta_s\right) \neq 0$ for each s. The higher the score, the higher the similarity between the TR service and the user profile; thus, the list of recommendations ranked in decreasing order based on the relevance scores can be returned.

4 LOOKER: implementation

LOOKER has been implemented to recommend travel-related services to foreign tourists visiting Tunisia. The first release of the application has been designed to manage two languages, i.e., English and French. The content-based filtering algorithm underlying the provided recommendations (detailed in Section 3) makes it possible to adapt the application to manage in the future different languages and geographic areas. The first LOOKER prototype has been developed with the TUNAV private company,[11] for the city of Tunis, and then it has been enriched by considering other three cities: Ariana, Sousse and Monastir. LOOKER has been developed for the Android mobile operating system, on top of Android Studio version 6 to support all smartphones running the Android "Marshmallow" version or higher.[12] Android is currently the largest mobile platform; it dominates the smartphone market with a share of 84%,[13] which allows to the proposed app to be exposed to the largest part of mobile users globally.

4.1 System architecture

LOOKER has been implemented as a *Rich Mobile Application* (RMA) [1], to leverage the hardware capabilities and specifically the GPS sensors of smartphones intended to run the spatio-temporal filtering module. The LOOKER's *client-server* architecture is illustrated in Fig. 1. Specifically, its components are organized in a *two-tier* client-server design in which each tier plays a specific role based on the individual roles of its modules.

In details, the *client side* includes a *graphical user interface* (GUI) and a *presentation logic* component. The presentation logic component handles and manages the user's interactions with the mobile application. It includes data validation, response to the user's actions, and communication between the GUI components. All information gathered by means of the GUI is sent to the LOOKER server. Here, data referring to travel-related services and to the user's preferences are stocked and treated.

At the *server side*, the *Google Places* and the *Google Maps* APIs as well as others APIs – detailed in Table 2 – are used to obtain details about TR services, such as opening hours, popular visiting times, reviews, and photos. The server includes two main components: (i) the *data repository* (DR), and (ii) the *recommendation engine*, which is in turn composed of the two distinct modules

[11] http://www.tunav.com/

[12] http://www.frandroid.com/tag/android-6-0-marshmallow/

[13] Source IDC, 2018: https://www.idc.com/promo/smartphone-market-share/os/

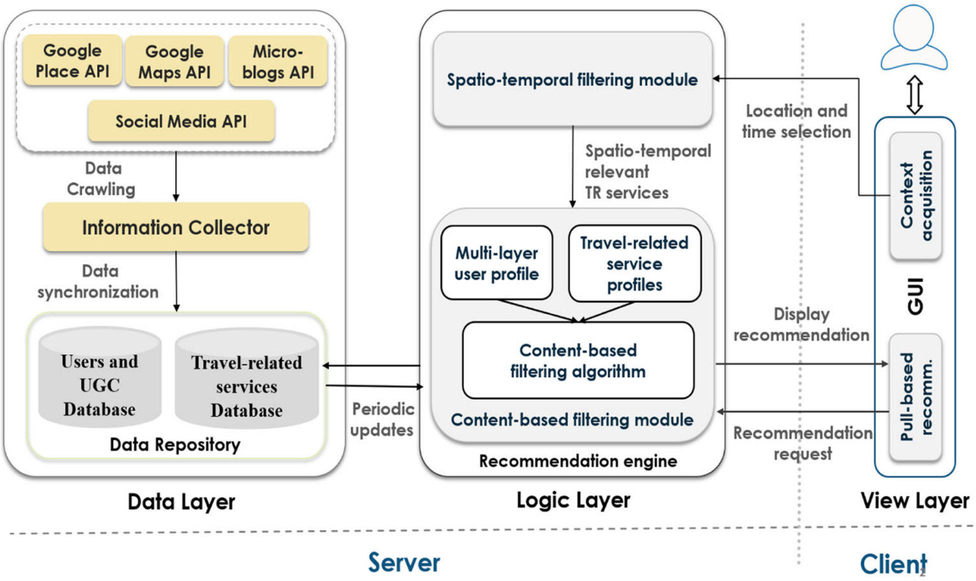


Fig. 1 The architecture of LOOKER

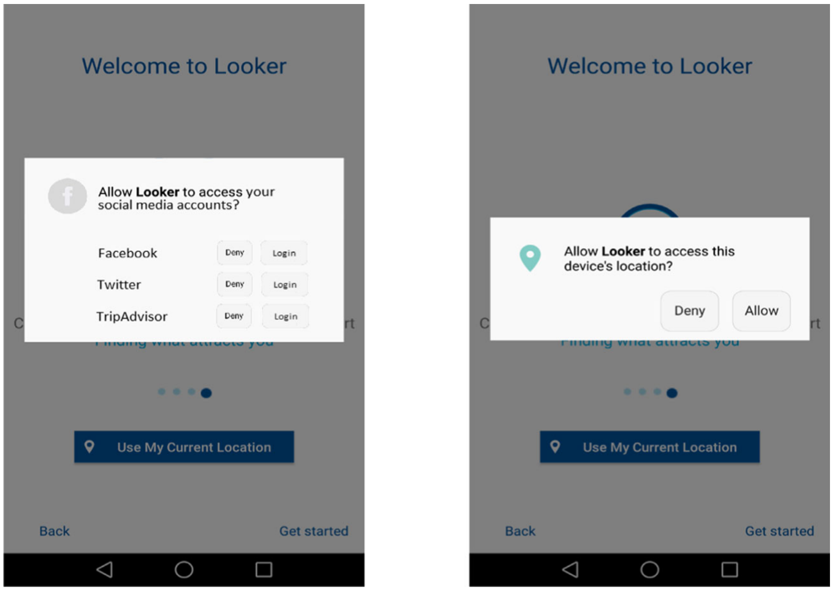
described in Section 3, i.e., the *spatio-temporal filtering module*, and the *content-based filtering module*. The DR acts as a back-end system for the recommendation process, and it is responsible for data persistence; data include in particular the collected user-generated content (i.e., on-line reviews, ratings) and TR services information.

A brief description of the global functioning of LOOKER is provided below, by emphasizing the interactions among the components of the architecture. Here, the different phases of the recommendation process are described. It is worth to be underlined that LOOKER provides *pull-based* recommendations, i.e., *reactive* recommendations that are the result of an explicit request coming from the user [18]. Specifically, for each target user who asks for recommendations:

- The *Android client* uses the location data from the GPS sensor of the user's smarthpone (e.g., longitude and latitude) to track her/his position; then, it communicates it to the LOOKER *server*. The user's location is updated at each user's recommendation request. In particular, each time a user opens the application and selects one of the TR-service categories s/he is interested in (i.e., *food*, *shopping*, *health*, or *attractions*), the location is updated (as illustrated in Section 4.2).
- The LOOKER *server*, by means of the *spatio-temporal filtering module*, selects a subset of TR services belonging to the TR-service category specified by the user based on her/his current location.
- The LOOKER *server* executes the recommendation process by means of the *content-based filtering module*. TR services are ranked by comparing them to the specific user profile layer corresponding to the TR-service category for which the recommendations have been requested by the user.
- The LOOKER *server* sends the ranked list of recommended travel-related services back to the *Android client*.

Table 2 Overview of the employed APIs

API	Description
Google Places API	It allows to get the current location, get periodic location updates, and addresses. It provides the estimated distance between two locations. It gives further details about services, including opening hours and popular visiting times, reviews, and photos.
Google Maps API	It allows to add maps to the application, and to customize them with additional content and imagery provided by LOOKER.
TripAdvisor API	It allows to get information for different points of interest. The information are locations, categories, reviews, and ratings, etc.
Facebook API	It allows the integration of the Facebook's Graph API, to get information like the user's name, likes, and interests.
Twitter API	It allows to get a collection of tweets posted by the target user.



Allow **Looker** to access your social media accounts?

Allow **Looker** to access this device's location?

(*a*) Social media accounts permission request. (*b*) GPS permission request.

Fig. 2 Required permissions

– Finally, the ranked list of TR services is shown to the user by means of the user interface provided by the *Android client* as illustrated in Figs. 3, 4, and 5.

4.2 Interface design

To get the highest level of usability, the first interaction that a user has with LOOKER is via *on-boarding* screens[14] illustrated in Fig. 3a, which introduce to a new user the LOOKER application, its operation, and some practical hints on how to better exploit the app. The purpose of the on-boarding screens is twofold: (i) to provide the user with a "learning by doing" methodology to interact with the application, and (ii) to show to the user the ease of use of LOOKER and how fun and useful it can be.

After the *on-boarding process*, the participants are requested to provide some permissions to LOOKER. Firstly, they must allow to LOOKER to have access to the user's different social media accounts, i.e., Twitter, Facebook, and TripAdvisor, as illustrated in Fig. 2a. This permission allows to the app to collect the user's textual UGC. Secondly, the user is required to give permission to LOOKER to identify her/his geographic location (i.e., using the latitude and longitude information obtained by the GPS), as shown in Fig. 2b.

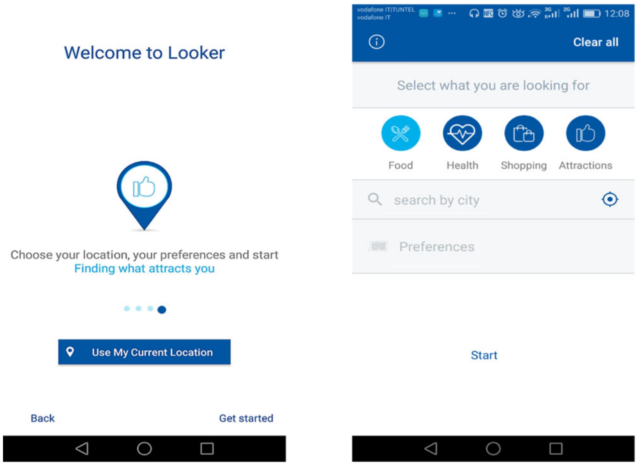
Once the permissions have been granted, the user must specify to which travel-related service category s/he is interested in, to receive touristic suggestions accordingly. S/he can select the chosen TR-service category by clicking on one of the top buttons, i.e., *Food*, *Health*, *Shopping*, and *Attractions*, as illustrated in Fig. 3b. Moreover, each time the user requests recommendations for a travel, s/he can specify and add other (optional) preferences in the form of keywords, which s/he thinks can be useful to enhance the results, by filling the *Preferences* field shown in Fig. 3b.

Figure 4 shows an example of a ranked list of recommendations in the *food* category (e.g., restaurants), which can be easily scrolled by the user. Recommended TR-services are provided in decreasing order, i.e., the highest is the service in the ranked list, the more suitable it should be to the user (according to the proposed CBF recommendation algorithm). By clicking on the MAP button (see Fig. 4), the user is able to visualize the set of recommended TR services on a map, and to verify their distance with respect her/his current position. Besides the visualization on the map, the user can see more details about each item in the ranked list by directly clicking on it. As illustrated in Fig. 5, the detailed view provides various information about the recommended TR service, such as name, address, description, Web site link, average rating given by other users, as well as the reviews written about it, and finally, the geolocation on the map.

5 LOOKER evaluation

The evaluation of recommender systems is a challenging task. In general, to evaluate the effectiveness of an RS, three possible strategies are reported in the literature [60]: *user studies*, *off-line experiments*, and *on-line experiments*.

[14] The *on-boarding process*, in mobile application, refers to the mechanism through which new users are informed and educated about the application. A strong on-boarding process is essential for the success of the mobile app [46].

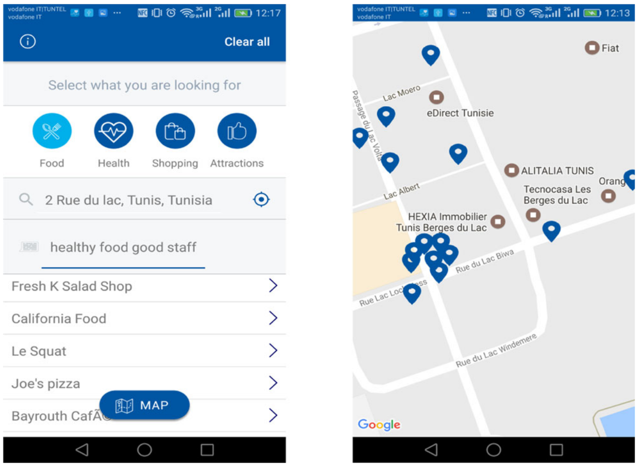


Choose your location, your preferences and start Finding what attracts you

(*a*) The on-boarding Interface.

(*b*) The main interface.

Fig. 3 The LOOKER mobile tourism recommender system



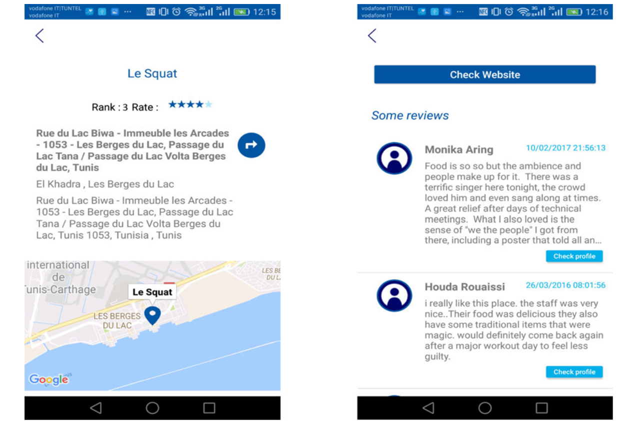
2 Rue du lac, Tunis, Tunisia

healthy food good staff

Set of food recommendations.

Map view of the results.

Fig. 4 Results of restaurants recommendations



Rue du Lac Biwa - Immeuble les Arcades - 1053 - Les Berges du Lac, Passage du Lac Tana / Passage du Lac Volta Berges du Lac, Tunis

El Khadra , Les Berges du Lac

Rue du Lac Biwa - Immeuble les Arcades - 1053 - Les Berges du Lac, Passage du Lac Tana / Passage du Lac Volta Berges du Lac, Tunis 1053, Tunisia , Tunis

Monika Aring 10/02/2017 21:56:13

Food is so so but the ambience and people make up for it. There was a terrific singer here tonight, the crowd loved him and even sang along at times. A great relief after days of technical meetings. What I also loved is the sense of "we the people" I got from there, including a poster that told all an...

Houda Rouaissi 26/03/2016 08:01:56

i really like this place. the staff was very nice..Their food was delicious they also have some traditional items that were magic. would definitely come back again after a major workout day to feel less guilty.

The detailed view of a recommended restaurant.

A set of reviews about the selected restaurant.

Fig. 5 LOOKER recommendation results details

In order to evaluate LOOKER, two user studies have been employed, the first one based on the use of two well-known questionnaires to measure the user's satisfaction and attitude towards the overall experience with the LOOKER mobile application, and the second one providing evidence about the user's satisfaction with respect to the obtained recommendations.

5.1 Evaluating the usability and the usefulness of the mobile application

The main motivation for performing a user study is its ability to take into account the user's experience [37] in interacting with a system. The first user study considered for evaluation purposes has been performed before the official release of the application. It is based on the use of two well-known questionnaires. Thanks to them it has been possible to assess both the *usability* of the system and its *usefulness* according to the users' judgments [55].

5.1.1 Selecting participants and main activities

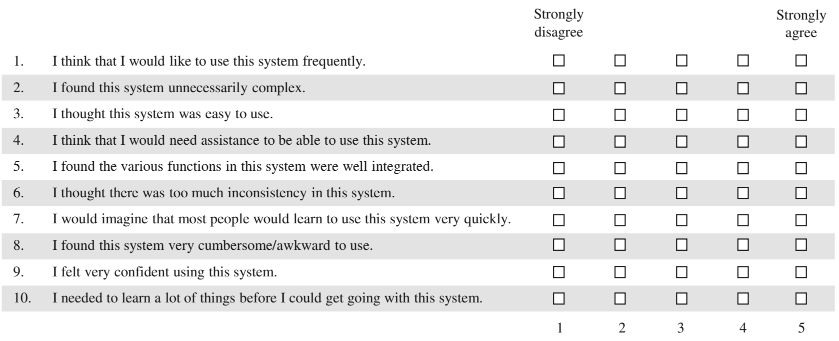
The user study has been conducted by recruiting, at a preliminary stage to the release of the application, a number of volunteers with different characteristics. According to Tullis et al. [64], to conduct an accurate user study [37], a group of at least 12/14 participants suitably selected is necessary. In the proposed study, in order to have a bigger set of users, 48 participants have been selected. Among the 48 people, 23 were female and 25 were male. Participants were teachers, graduate, and undergraduate students from three Tunisian universities (ISG Sousse,[15] ESC Manouba,[16] and IHEC Carthage[17]), computer science engineers (Android developers) who represent both "experts" and simple travelers not related to the academic environment. In general, they were young smartphone users who like to travel and whose age ranged between 21 and 33 years.

The participants were first requested to provide background information about themselves, such as demographic information and their knowledge about tourism mobile RS and mobile applications in general. Then, they were also asked to provide reviews for TR services that they already experienced, which was necessary to build their user profiles. Users were required to provide at least 60 reviews in English for TR services belonging to the four different categories (i.e., *food*, *shopping*, *health*, *attractions*). Not being real users of the application, this phase was necessary for

[15] http://www.isgs.rnu.tn/

[16] http://www.esct.rnu.tn/site/

[17] http://www.ihec.rnu.tn/



		Strongly disagree				Strongly agree
1.	I think that I would like to use this system frequently.	□	□	□	□	□
2.	I found this system unnecessarily complex.	□	□	□	□	□
3.	I thought this system was easy to use.	□	□	□	□	□
4.	I think that I would need assistance to be able to use this system.	□	□	□	□	□
5.	I found the various functions in this system were well integrated.	□	□	□	□	□
6.	I thought there was too much inconsistency in this system.	□	□	□	□	□
7.	I would imagine that most people would learn to use this system very quickly.	□	□	□	□	□
8.	I found this system very cumbersome/awkward to use.	□	□	□	□	□
9.	I felt very confident using this system.	□	□	□	□	□
10.	I needed to learn a lot of things before I could get going with this system.	□	□	□	□	□
		1	2	3	4	5

Fig. 6 The system usability scale questionnaire

the correct operation and evaluation of the system. Participants were requested to provide reviews as much as possible of quality, avoiding to write too short reviews or to copy reviews already written. It has been demonstrated in the literature that, if properly written, genuine reviews and ah-hoc written reviews are almost indistinguishable [69].

Then, the LOOKER application was briefly introduced to the participants, and the purpose of the user study was presented. After introducing the participants to the task, they installed the LOOKER application on their smartphones. They were asked to test LOOKER for 15 days in different locations; after having performed this phase, participants were asked to fill two popular questionnaires that are often employed for the evaluation of different kinds of applications, i.e., the *System Usability Scale* (SUS) questionnaire, and the *Computer System Usability Questionnaire* (CSUQ). Specifically, the SUS questionnaire allows to measure the usability of a system in general, while the CSUQ allows to assess four aspects of the usability of a system: interface quality, information quality, system usefulness and overall satisfaction. Both questionnaires have a higher accuracy with an increasing sample size with respect to other usability questionnaires such as the Questionnaire for User Interface Satisfaction (QUIS) [64]. Details on both questionnaires are provided in the following sections.

5.1.2 SUS questionnaire

The *System Usability Scale* (SUS) questionnaire [15] represents a simple and reliable tool for a system's usability evaluation. As illustrated in Fig. 6, it is a short questionnaire that includes 10 questions where participants indicate their rate of approval on a 5-point Likert scale, whose values corresponds to 1: Strongly disagree, 2: Disagree, 3: Neutral, 4: Agree, and 5: Strongly agree.

The SUS questionnaire has been tested throughout almost 30 years of use, and has proven to be an effective method for evaluating the usability of systems. SUS yields a single number representing a composite measure of the overall usability of the system under analysis. For further details on how calculate the SUS score, please refer to [16].

Figure 7 illustrates the correspondence between the scores that the SUS questionnaire can produce (on a [0 − 100] range) and the values of other scales proposed in [6]. This allows to evaluate in a clear way the overall score obtained via the SUS questionnaire after having used LOOKER. In particular, the global score obtained by aggregating the ratings provided by our group of participants after trying the LOOKER application during 15 days is 82.9. According to the correspondence between the SUS scores and other scales (see Fig. 7), LOOKER has an "excellent" level of usability.

The SUS questionnaire to evaluate the impact of the on-boarding process As previously illustrated, to improve the usability of the LOOKER system, an *on-boarding process* [46] has been implemented, with the aim of helping users in understanding the key functionalities of LOOKER and to improve the user's first impression about the application. To evaluate the impact of the on-boarding process on the

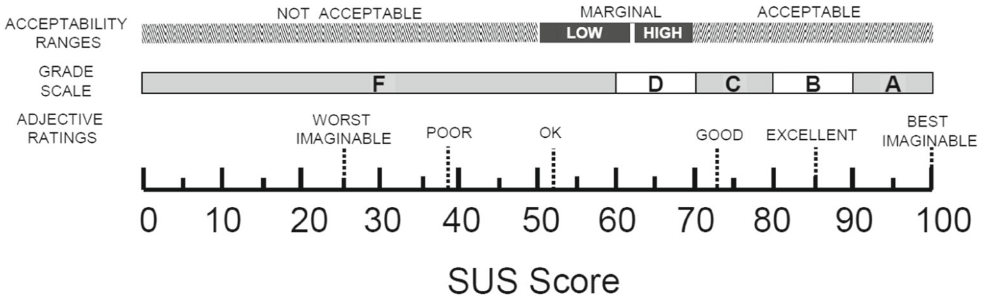


Fig. 7 A comparison between the SUS and other grade rankings reproduced based on [6]. Reprinted with permission

LOOKER's usability, participants have been divided into two groups. The first group was composed of 24 users and filled the SUS questionnaire after using the application without the *on-boarding process*. The rest of the participants assigned their score values in the SUS questionnaire after using a LOOKER version improved with on-boarding screens. Later, the global SUS scores obtained for each of the two groups of users after using the proposed application with and without on-boarding screens have been computed. The SUS score was 80.3 without on-boarding, while it achieved the value of 82.9 with the on-boarding process, with a growth of 3.23%. These results demonstrate that a solid on-boarding process allows to improve the overall system usability.

5.1.3 Computer system usability questionnaire

The *Computer System Usability Questionnaire* (CSUQ) [41] has been employed to measure the user satisfaction in using LOOKER under different aspects. The CSUQ consists of 19 questions, for which users were required to provide ratings on a 7-point Likert scale. As illustrated in Fig. 8, the possible ratings range from 1: "Strongly disagree" to 7: "Strongly agree". The CSUQ questions can be classified into three categories (or sub-scales):

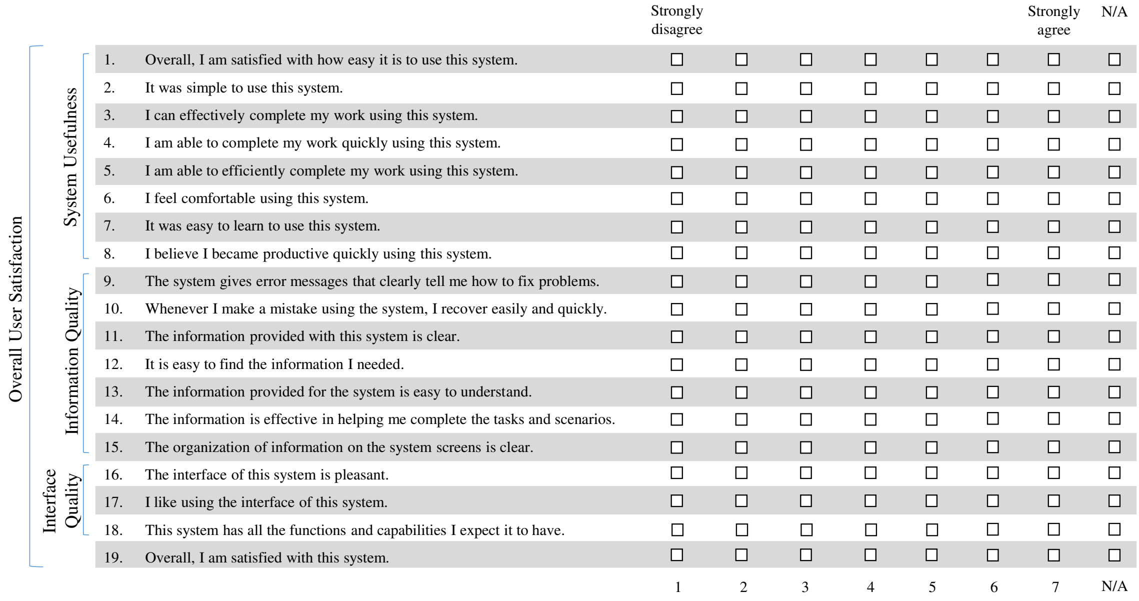
- *System usefulness*: questions 1–8 report the system usefulness;
- *Information quality*: questions 9–15 evaluate the user's perceived satisfaction with respect to the quality of the information associated with the system (e.g., information clarity);
- *Interface quality*: questions 16–19 allow to assess the interface quality.

The CSUQ scores have been first computed with respect to the three sub-scales: the *System Usefulness*, which corresponds to 5.52, the *Information Quality*, equal to 5.53, and the *Interface Quality*, equal to 5.23. All these scores are averaged on a 7-point scale, where high scores are better than low scores (see the anchors used in the 7-point scales, as it emerges from Fig. 8). The global CSUQ score is equal to 5.25 computed on the 7-point scale, which implies a high level of usability. These scores have been obtained as described in [41].

The global score has also been normalized on a $[0-100]$ scale, to compare it with the result produced by the SUS score. The obtained result, equal to 74.4, indicates that the LOOKER app is generally perceived as "good" (see Fig. 7). To a greater level of detail, the proposed approach has been particularly appreciated with respect to *Information Quality* (the normalized score for this sub-scale is 76.75).

5.1.4 SUS-CSUQ scores and demographic factors

In Table 3, the SUS and the CSUQ scores are presented with respect to different demographic factors. Specifically, the *gender*, *educational background*, and *occupation* of users are considered. The SUS and CSUQ scores for each of these demographic aspects are reported, along with their standard deviation (SD) values. As it emerges from the table, for the specific group of users considered, the mobile app has been particularly appreciated by women, and in general by graduate people.



				Strongly disagree						Strongly agree	N/A
Overall User Satisfaction	System Usefulness	1.	Overall, I am satisfied with how easy it is to use this system.	□	□	□	□	□	□	□	□
		2.	It was simple to use this system.	□	□	□	□	□	□	□	□
		3.	I can effectively complete my work using this system.	□	□	□	□	□	□	□	□
		4.	I am able to complete my work quickly using this system.	□	□	□	□	□	□	□	□
		5.	I am able to efficiently complete my work using this system.	□	□	□	□	□	□	□	□
		6.	I feel comfortable using this system.	□	□	□	□	□	□	□	□
		7.	It was easy to learn to use this system.	□	□	□	□	□	□	□	□
		8.	I believe I became productive quickly using this system.	□	□	□	□	□	□	□	□
	Information Quality	9.	The system gives error messages that clearly tell me how to fix problems.	□	□	□	□	□	□	□	□
		10.	Whenever I make a mistake using the system, I recover easily and quickly.	□	□	□	□	□	□	□	□
		11.	The information provided with this system is clear.	□	□	□	□	□	□	□	□
		12.	It is easy to find the information I needed.	□	□	□	□	□	□	□	□
		13.	The information provided for the system is easy to understand.	□	□	□	□	□	□	□	□
		14.	The information is effective in helping me complete the tasks and scenarios.	□	□	□	□	□	□	□	□
		15.	The organization of information on the system screens is clear.	□	□	□	□	□	□	□	□
	Interface Quality	16.	The interface of this system is pleasant.	□	□	□	□	□	□	□	□
		17.	I like using the interface of this system.	□	□	□	□	□	□	□	□
		18.	This system has all the functions and capabilities I expect it to have.	□	□	□	□	□	□	□	□
		19.	Overall, I am satisfied with this system.	□	□	□	□	□	□	□	□
				1	2	3	4	5	6	7	N/A

Fig. 8 The computer system usability questionnaire

Table 3 A summary of mean values of CSUQ and SUS scores for different categories of participants

Demographic factor	Response	CSUQ score		SUS score	
		Mean	SD	Mean	SD
Gender	Male	74.38	6.85	85.30	8.04
	Female	77.67	7.84	85.62	8.18
Education background	Graduate	79.17	6.15	86.66	8.74
	Undergraduate	73.54	8.13	84.37	7.34
Occupation	Teacher	72.77	9.14	84.61	8.65
	Software Engineer	71.03	7.62	84.16	7.01
	Others	72.29	6.19	86.15	8.13

5.2 Evaluating recommendations

Another user study has been put in place to evaluate the LOOKER's recommendation algorithm, this time with real users, after the release of the application. The ranked lists obtained by the LOOKER's recommendation algorithm, denoted as CACB (*context-aware, content-based*), have been compared against those produced by a state-of-the-art approach, by means of an A/B testing and suitable evaluation metrics.

5.2.1 Description of the procedure

For the purpose of the user study, a Web-centric testing framework for the tourism domain has been developed, which can be easily configured to facilitate the execution of controlled studies. The testing framework is split into two sessions that rely on two different recommendation algorithms: (i) the CACB algorithm described in Section 3, and (ii) the baseline algorithm that will be detailed in Section 5.2.3. For both algorithms, the user interface was the same, in order to avoid the potential bias deriving from the system's interface influence on the users' behavior. In particular, the A/B testing methodology (also known as *split testing*) [72] has been considered, because it has been extensively employed in many domains [4, 38] including recommender systems [24] and personalized search [31]. This testing methodology has been used to compare the CACB and the baseline algorithms against each other, to determine which one performs better (in terms of recommendation).

For the A/B testing, a new group of 120 users has been considered. They have been recruited among real users of the application, and have been selected in a way to maximize their differences in terms of age range, gender, educational background, occupation. Being real users with a profile on LOOKER, it has not been necessary in this second user study to ask them to explicitly provide hand-written reviews. The 120 users have been split into A/B test groups randomly, i.e., 60 users in each group. The group A has received recommendations by the CACB algorithm, while the group B by the baseline algorithm. Globally, 5,227 recommendations were delivered to the 120 users.

The interaction with the testing framework starts with a sign-up process; after that, the testing process takes place as follows:

1. At each interaction, a user is asked to imagine being in a given contextual scenario, as it will be illustrated in detail in Section 5.2.2.
2. On the basis of this scenario, a recommendation list is generated and provided to the user, containing up to 20 TR services.
3. The user is asked to browse the TR services in the recommendation list. For each service, the user can see some details such as the address, the map view, the ratings, the Web site (if it is available), and the reviews provided by other users.
4. Then, the user is asked to rate each TR service within the recommendation list on a 5-star scale, either based on a prior knowledge s/he had of the recommended service, or based on the service's details s/he has analyzed in the previous step. These ratings are used as explicit relevance feedbacks for evaluating the recommendations provided by both algorithms (the one proposed and the baseline) as explained in Sections 5.2.4, and 5.2.5.

5.2.2 Simulating the contextual scenario

Before being able to recommend items to users based on the context, contextual information has to be taken into account. To this aim, the testing framework generates specific mobile scenarios at each run. In fact, users were asked to imagine

Table 4 A/B test statistics

	CACB under group A (%)	CSLIM under group B (%)
5/5 ratings	34.06	26.52
4/5 ratings	19.91	16.38
3/5 ratings	43.67	42.71
< 3/5 ratings	2.36	14.39

the scenario as their current context. The scenario is visually displayed to the user throughout the user study at the top of the screen, whereby:

- The *location* can be Tunis, Ariana, Sousse, Monastir.
- The *location type* can be at work, at home, at university, outside.
- The *time* can be morning (7:00-11:00), noon (11:00-14:00), afternoon (14:00-18:00), evening (18:00-21:00), Night (21:00-Next day 7:00).
- The *companionship* can be with my friends, with my family, with my partner.

5.2.3 The baseline algorithm

The proposed CACB algorithm has been evaluated with respect to a state-of-the-art context-aware recommendation algorithm, i.e., the *Contextual SLIM Recommendation Algorithm* (CSLIM) [78]. The CSLIM extends the *Sparse LInear Method* (SLIM) algorithm [48] by incorporating contextual conditions, and assumes that the user's preferences vary from context to context for the same item. Specifically, CSLIM estimates a *relevance score* $\hat{r}_{u,i,c}$ for a user u on an item i in a context c as follows:

$$\hat{r}_{u,i,c} = \sum_{h,j,h\neq j}^{N} r_{u,j,c} \cdot w_{h,j} \,, \quad (5)$$

where the $r_{u,j,c}$ values are *contextual ratings* that u provided on other items j in the same context c, the $w_{h,j}$ values are item-item *similarity coefficients*, computed between items h and j under the same context, and N is the total number of items. Since contextual ratings $r_{u,j,c}$ are usually sparse, i.e., multiple ratings for the same user u in the same context c might be not always available, it is possible to estimate the user u's rating on an item j in a context c based on the user's non-contextual ratings on this item, and on the aggregated contextual rating deviations (CRD)s, i.e., the rating deviations in different contextual conditions (e.g., a restaurant meant for lunch breaks may obtain a 5-star rating in a "weekday afternoon," while a 2-star rating in a "weekend evening"). Different CSLIM models can be built based on how CRDs are estimated. For evaluation purposes, the CSLIM-I model detailed in [78] has been considered in this paper, and, in particular, its implementation provided by the CARSKit tool [79].

The choice of this algorithm as a baseline has been undertaken since, according to [78], the CSLIM algorithm outperforms splitting approaches for context-aware recommendation [77], in particular with respect to two datasets referred to the "food" [49] and the "music" [5] categories.

5.2.4 Simple statistical results

By the proposed user study, and the considered A/B testing, it is possible to provide some simple statistics with respect to the explicit relevance feedbacks provided by users in each group, i.e., with respect to the two distinct recommendation algorithms. As it is summarized in Table 4, participants in the group A (who tested the proposed CACB recommendation algorithm) provided around 55% of their ratings in the 4–5-star interval (on a 5-star scale), while participants in the group B (who tested the CSLIM baseline) provided around 43% of their ratings in the same interval.

Figure 9 gives more details about the distribution of the ratings provided by participants with respect to the two

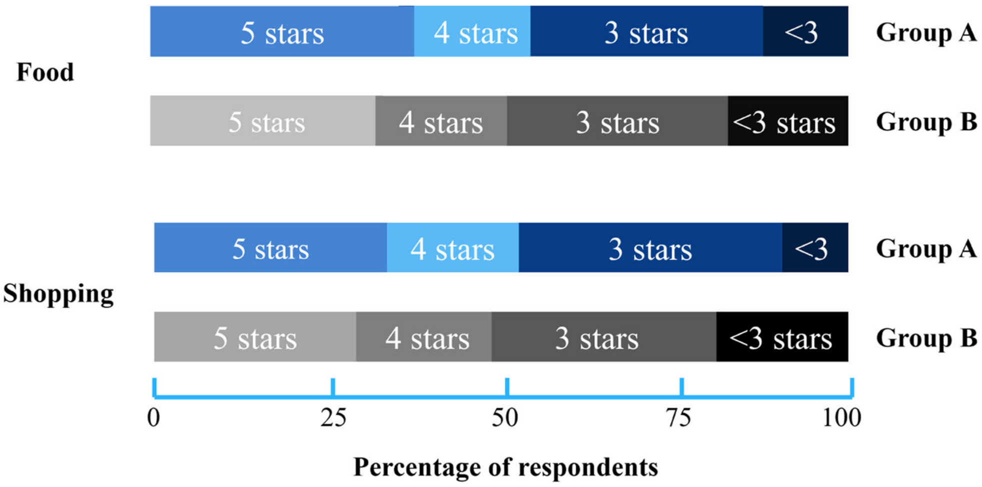


Fig. 9 Relevance feedbacks provided by participants from the group A and B

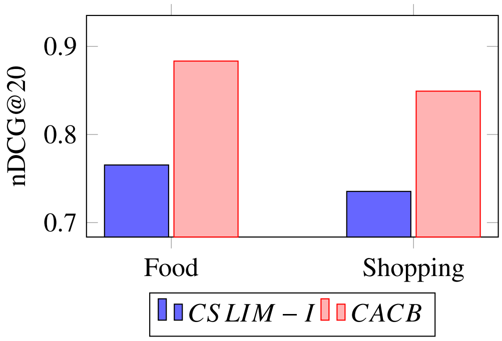


Fig. 10 NDCG@20 results of the evaluated recommendations algorithms

travel-related service categories, i.e., *food* and *shopping*, which have been considered in the evaluations. More than 95% of the users of the group A provided ratings with a value above 3 stars. Concerning the members of the group B, the percentage of users having provided ratings higher than 3 stars is 80%.

5.2.5 Ranking quality

In the A/B testing, users receive recommendation lists and evaluate each TR service in the list with an explicit rating on a 5-star scale, by considering also some contextual aspects.[18] In the evaluation process, these ratings are considered as *graded relevance judgments* which measure the *gain*, or *usefulness* that each item has in the result list based on its position in the list. Highly relevant documents are more useful than marginally relevant documents; for this reason, the lower the ranked position of a relevant document, the less useful it is for the user. For this reason, the gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks. Items that are not explicitly evaluated by users are considered as not relevant, and have a graded relevance judgment equal to zero [30].

This way, it is possible to employ the *normalized Discounted Cumulative Gain* (nDCG) [34] metric to evaluate the *ranking quality* of the two considered recommendation algorithms (the CACB and the CSLIM algorithms). Formally, assuming that each user u has a "gain" g_{u,i_j} if the item i is presented to her/him at rank j, the average *Discounted Cumulative Gain* (DCG) over all users for a list of m items is defined as:

$$\mathrm{DCG} = \frac{1}{n}\sum_{u=1}^{n}\sum_{j=1}^{m}\frac{g_{u,i_j}}{\log_2(j+1)}, \tag{6}$$

Table 5 Statistics on data collected

	Food	Shopping
Average number of reviews	42.5	37.2
#Users	120	
#Items	5,227	

where n is the total number of users.

nDCG is the normalized version of DCG, given by:

$$\mathrm{nDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}, \tag{7}$$

where IDCG is the ideal DCG, obtained by sorting all relevant documents in the corpus by their relative relevance, producing the maximum possible DCG [20].

The cut-off version of nDCG, i.e., nDCG@k, set the discount to be zero for ranks larger than k. In this work, we consider nDCG@20; Fig. 10 reports the nDCG@20 values for the CACB and the CSLIM algorithms with respect to the categories *food* and *shopping*. We can observe that CACB outperforms CSLIM in both categories, but it has the largest gain with respect to CSLIM in the *food* category ($> 20\%$) and a lower gain in the *shopping* category (15%).

The best performance achieved in the *food* category can be explained by the higher number of reviews available for this category, as illustrated in Table 5. This represents an indication of the fact that, when a sufficient amount of UGC is available, this can constitute an effective source for identifying the user's preferences; therefore, it is possible to reduce the limited content analysis problem, and to improve the user's satisfaction with respect to the recommendations s/he receives.

6 Conclusion and further research

In this paper, the mobile app LOOKER has been presented, which is a recommender system for the tourism domain. LOOKER is based on a content-based filtering algorithm that exploits the content generated by the user to infer her/his interests and preferences. Specifically, the language employed by the user in her/his tourism-related social media posts is used to build a user profile, modeled via language models. Recommendations are then provided to the user by considering the similarity between her/his user profile and the formal representation of travel-related services, which is still also modeled as a statistical language model. LOOKER makes use of some basic contextual information in the recommendation process, i.e., location and time; to this aim the contextual pre-filtering paradigm has been adopted. This means that the considered contextual information

[18] This strategy aims at tackling a major issue in evaluating CARS, i.e., the lack of context-enriched datasets [30].

is used to filter out irrelevant items, and the proposed content-based filtering algorithm is employed to generate recommendations on the remaining items.

The mobile application and the underlying recommendation model have been evaluated by employing two different user studies. A first user study, aimed to test the usability of the proposed application with a set of users in four big cities in Tunisia, was based on the use of two popular questionnaires (i.e., SUS and CSUQ). A second user study has allowed to quantitatively evaluate the effectiveness of the recommendations produced by the proposed system.

Both the app usability and the provided recommendations have been judged as satisfactory by users. Despite this, some aspects that have not been sufficiently taken into account in the development of this first version of the application will have to be addressed in the future. First of all, additional aspects of the user's context will be taken into consideration. Multiple contextual dimensions can be captured by a mobile app, which can be useful in providing personalized recommendations, other than just location and time. Contextual ratings could be incorporated in the content-based recommendation model, in order to capture the circumstances in which the opinion (in the form of an on-line review) was made, e.g., detecting if the opinion on a restaurant has been provided at lunch-time or at dinner-time, during a weekday or a weekend, during a group meal or a couple meal.

In addition to this, also the content-based filtering recommendation algorithm could be improved, by considering varying levels of satisfaction expressed by positive reviews. In fact, there are some signals that could be employed to differentiate reviews based for example on the ratings provided or on the sentiment that emerges from the reviews.

Furthermore, concerning evaluations, in the first of the performed user studies some volunteers have been selected before the release of the application. Given some limitations in the recruitment process, only a small number of users poorly diversified in terms of age and educational background has been considered. To better evaluate the usability of the application, another study that takes into consideration this problem must be be carried out. This issue has not affected the second performed user study; in fact, in this case, to evaluate the effectiveness of the provided recommendations, diversified real users (i.e., already utilizing the mobile application) were recruited.

In future works, some other issues not tackled in this paper, related to the analysis of the textual content of UGC, will be addressed, such as polysemy and synonymy [22]. Furthermore, term-dependency issues could be tackled by using bigrams/trigrams language models to represent the user profile. Also, opinion mining techniques, which require a deeper analysis of the sentiment lexicon employed in social media posts could be applied. Finally, some aspects related to the privacy of users and the confidentiality of their contents could be investigated [27].

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Abolfazli S, Sanaei Z, Gani A, Xia F, Yang LT (2014) Rich mobile applications: genesis, taxonomy, and open issues. J Netw Comput Appl 40:345–362
2. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
3. Adomavicius G, Tuzhilin A (2015) Context-aware recommender systems. In: Recommender systems handbook. Springer, pp 191–226
4. Andersen E, Liu YE, Snider R, Szeto R, Popović Z (2011) Placing a value on aesthetics in online casual games. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 1275–1278
5. Baltrunas L et al (2011) InCarMusic: Context-aware music recommendations in a car. In: Huemer C, Setzer T (eds) E-commerce and Web technologies. EC-Web 2011. Lecture notes in business information processing, vol 85. Springer, Berlin
6. Bangor A, Kortum P, Miller J (2009) Determining what individual sus scores mean: adding an adjective rating scale. J Usability Stud 4(3):114–123
7. Batet M, Moreno A, Sánchez D, Isern D, Valls A (2012) Turist@: agent-based personalised recommendation of tourist activities. Expert Syst Appl 39(8):7319–7329
8. Belbachir F, Boughanem M, Missen MMS (2014) Probabilistic opinion models based on subjective sources. In: Proceedings of the 29th annual ACM symposium on applied computing. ACM, pp 925–926
9. Bettini C, Brdiczka O, Henricksen K, Indulska J, Nicklas D, Ranganathan A, Riboni D (2010) A survey of context modelling and reasoning techniques. Pervasive Mob Comput 6(2):161–180
10. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl-Based Syst 46:109–132
11. Borras J, Moreno A, Valls A (2014) Intelligent tourism recommender systems: a survey. Expert Syst Appl 41(16):7370–7389
12. Bouwman H, Carlsson C, Lopez-Nicolas C, Mckenna B, Molina-Castillo F, Tuunanen T, Walden P (2011) Mobile travel services: the effect of moderating context factors. Inform Technol Tourism 13(2):57–74
13. Braunhofer M, Kaminskas M, Ricci F (2011) Recommending music for places of interest in a mobile travel guide. In: Proceedings of the fifth ACM conference on recommender systems. ACM, pp 253–256
14. Braunhofer M, Ricci F et al (2017) Selective contextual information acquisition in travel recommender systems. Inform Technol Tourism 17(1):5–29
15. Brooke J (2013) SUS: a retrospective. J Usability Stud 8(2):29–40
16. Brooke J et al (1996) SUS - A quick and dirty usability scale. Usability Eval Industry 189(194):4–7
17. Cantoni L, Saldaña MTL (2016) Mobile systems for tourism. Inform Technol Tourism 16(2):149–151
18. del Carmen Rodríguez-Hernández M, Ilarri S (2016) Pull-based recommendations in mobile environments. Comput Standards Int 44:185–204

19. Cenamor I, de la Rosa T, Núñez S, Borrajo D (2017) Planning for tourism routes using social networks. Expert Syst Appl 69:1–9
20. Clarke CL, Kolla M, Cormack GV, Vechtomova O, Ashkan A, Büttcher S, MacKinnon I (2008) Novelty and diversity in information retrieval evaluation. In: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval. pp. 659–666. SIGIR '08. ACM, New York
21. Colomo-Palacios R, García-peñalvo FJ, Stantchev V, Misra S (2017) Towards a social and context-aware mobile recommendation system for tourism. Pervasive Mobile Comput 38:505–515
22. Croft WB, Metzler D, Strohman T (2010) Search engines: information retrieval in practice, vol 283. Addison-Wesley, Reading
23. d'Aveni RA, Gunther RE (1995) Hypercompetitive rivalries: competing in highly dynamic environments Free Pr
24. Davidson J, Liebald B, Liu J, Nandy P, Van Vleet T, Gargi U, Gupta S, He Y, Lambert M, Livingston B et al (2010) The youtube video recommendation system. In: Proceedings of the fourth ACM conference on recommender systems. ACM, pp 293–296
25. Dey AK (2001) Understanding and using context. Personal Ubiquitous Comput 5(1):4–7
26. Felfernig A, Gordea S, Jannach D, Teppan E, Zanker M (2007) A short survey of recommendation technologies in travel and tourism. OEGAI J 25(7):17–22
27. Ferrari E, Viviani M (2013) Privacy in social collaboration. In: Handbook of human computation. Springer, pp 857–878
28. Gavalas D, Kenteris M (2011) A Web-based pervasive recommendation system for mobile tourist guides. Pers Ubiquit Comput 15(7):759–770
29. Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) Mobile recommender systems in tourism. J Netw Comput Appl 39:319–333
30. Gunawardana A, Shani G (2015) Evaluating recommender systems. In: Recommender systems handbook. Springer, pp 265–308
31. Hannak A, Sapiezynski P, Molavi Kakhki A, Krishnamurthy B, Lazer D, Mislove A, Wilson C (2013) Measuring personalization of Web search. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 527–538
32. Henricksen K, Indulska J (2006) Developing context-aware pervasive computing applications: models and approach. Pervasive Mobile Comput 2(1):37–64
33. Huang H (2016) Context-aware location recommendation using geotagged photos in social media. ISPRS Int J Geo-Information 5(11):195
34. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) 20(4):422–446
35. Kennedy-Eden H, Gretzel U (2012) A taxonomy of mobile applications in tourism. E-review Tourism Res 10(2):47–50
36. Kenteris M, Gavalas D, Economou D (2009) An innovative mobile electronic tourist guide application. Personal Ubiquitous Comput 13(2):103–118
37. Knijnenburg BP, Willemsen MC (2015) Evaluating recommender systems with user experiments. In: Recommender systems handbook. Springer, pp 309–352
38. Kohavi R, Henne RM, Sommerfield D (2007) Practical guide to controlled experiments on the Web: listen to your customers not to the hippo. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 959–967
39. Kullback S, Leibler RA (1951) On information and sufficiency. Annal Math Stat 22(1):79–86
40. Levandoski JJ, Sarwat M, Eldawy A, Mokbel MF (2012) Lars: a location-aware recommender system. In: 2012 IEEE 28th International Conference on Data Engineering (ICDE). IEEE, pp 450–461
41. Lewis JR (1995) Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. Int J Human-Comput Int 7(1):57–78
42. Loh S, Lorenzi F, Saldaña R, Licthnow D (2003) A tourism recommender system based on collaboration and text analysis. Inform Technol Tourism 6(3):157–165
43. Lops P, De Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Recommender systems handbook. Springer, pp 73–105
44. Lucas JP, Luz N, Moreno MN, Anacleto R, Figueiredo AA, Martins C (2013) A hybrid recommendation approach for a tourism system. Expert Syst Appl 40(9):3532–3550
45. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the fourth ACM international conference on Web search and data mining. ACM, pp 287–296
46. (2018) Material Design: Onboarding. https://material.io/design/communication/onboarding.html, [Online; accessed 31-Oct-2018]
47. Missaoui S, Viviani M, Faiz R, Pasi G (2017) A language modeling approach for the recommendation of tourism-related services. In: Proceedings of the 32st annual ACM symposium on applied computing. ACM, pp 1075–1076
48. Ning X, Karypis G (2011) Slim: sparse linear methods for top-n recommender systems. In: 11th IEEE International Conference on Data Mining (ICDM). IEEE, pp 497–506
49. Ono C, Takishima Y, Motomura Y, Asoh H (2009) Context-aware preference model based on a study of difference between real and supposed situation data. In: Houben GJ, McCalla G, Pianesi F, Zancanaro M (eds) User modeling, adaptation, and personalization. UMAP 2009. Lecture notes in computer science, vol 5535. Springer, Berlin
50. Panniello U, Gorgoglione M (2011) A contextual modeling approach to context-aware recommender systems. In: Proceedings of the 3rd workshop on context-aware recommender systems
51. Panniello U, Tuzhilin A, Gorgoglione M (2014) Comparing context-aware recommender systems in terms of accuracy and diversity. User Model User-Adap Inter 24(1-2):35–65
52. Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. Expert Syst Appl 39(11):10059–10072
53. Ponte JM, Croft WB (1998) A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 275–281
54. Poslad S, Laamanen H, Malaka R, Nick A, Buckle P, Zipl A (2001) Crumpet: creation of user-friendly mobile services personalised for tourism. In: Proceedings of the second international conference on 3G mobile communication technologies. IET
55. Rashid U, Viviani M, Pasi G (2016) A graph-based approach for visualizing and exploring a multimedia search result space. Inf Sci 370:303–322
56. Ricci F (2010) Mobile recommender systems. Inform Technol Tourism 12(3):205–231
57. Ricci F, Nguyen QN (2007) Acquiring and revising preferences in a critique-based mobile recommender system. IEEE Intell Syst 22(3):22–29
58. Savage NS, Baranski M, Chavez NE, Höllerer T (2012) I'm feeling loco: a location based context aware recommendation system. In: Advances in location-based services. Springer, pp 37–54
59. Sebastia L, Garcia I, Onaindia E, Guzman C (2009) e-tourism: a tourist recommendation and planning application. Int J Artif Intell Tool 18(05):717–738
60. Shani G, Gunawardana A (2011) Evaluating recommendation systems. In: Recommender systems handbook. Springer, pp 257–297

61. Stanciu O, Ṫichindelean M (2010) Consumer behavior in the different sectors of tourism. Stud in Business Econ 5(3):277–285
62. Telfer DJ, Sharpley R (2015) Tourism and development in the developing world Routledge
63. Tsai CY, Chung SH (2012) A personalized route recommendation service for theme parks using rfid information and tourist behavior. Decis Support Syst 52(2):514–527
64. Tullis TS, Stetson JN (2004) A comparison of questionnaires for assessing website usability. In: Usability professional association conference, pp 1–12
65. Tumas G, Ricci F (2009) Personalized mobile city transport advisory system. Inform Commun Technol Tourism 2009:173–183
66. Tussyadiah IP, Zach FJ (2012) The role of geo-based technology in place experiences. Ann Tour Res 39(2):780–800
67. Vansteenwegen P, Souffriau W, Berghe GV, Van Oudheusden D (2011) The city trip planner: an expert system for tourists. Expert Syst Appl 38(6):6540–6546
68. Villegas NM, Sánchez C, Díaz-cely J, Tamura G (2018) Characterizing context-aware recommender systems: a systematic literature review. Knowl-Based Syst 140:173–200
69. Viviani M, Pasi G (2017) Credibility in social media: opinions, news, and health information - a survey. Wiley Interdisciplinary Rev: Data Mining Knowl Discovery 7(5):e1209
70. Viviani M, Pasi G (2017) Quantifier guided aggregation for the veracity assessment of online reviews. Int J Intell Syst 32(5):481–501
71. Wang D, Xiang Z, Fesenmaier DR (2016) Smartphone use in everyday life and travel. J Travel Res 55(1):52–63
72. Wilson JD, Uminsky DT (2017) The power of A/B testing under interference. arXiv:1710.03855
73. Yang WS, Hwang SY (2013) itravel: a recommender system in mobile peer-to-peer environment. J Syst Softw 86(1):12–20
74. Younus A, O'Riordan C, Pasi G (2014) A language modeling approach to personalized search based on users' microblog behavior. In: European conference on information retrieval. Springer, pp 727–732
75. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware point-of-interest recommendation. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 363–372
76. Zhai C, Lafferty J (2004) A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS) 22(2):179–214
77. Zheng Y, Burke R, Mobasher B (2014) Splitting approaches for context-aware recommendation: an empirical study. In: Proceedings of the 29th annual ACM symposium on applied computing. ACM, pp 274–279
78. Zheng Y, Mobasher B, Burke R (2014) Cslim: contextual slim recommendation algorithms. In: Proceedings of the 8th ACM conference on recommender systems. ACM, pp 301–304
79. Zheng Y, Mobasher B, Burke R (2015) Carskit: a java-based context-aware recommendation engine. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, pp 1668–1671
80. Zhu Q, Wang S, Cheng B, Sun Q, Yang F, Chang RN (2018) Context-aware group recommendation for point-of-interests. IEEE Access 6:12129–12144

Affiliations

Sondess Missaoui[1,2] · Faten Kassem[3] · Marco Viviani[1] · Alessandra Agostini[1] · Rim Faiz[4] · Gabriella Pasi[1]

Sondess Missaoui
s.missaoui@disco.unimib.it

Faten Kassem
faten.kassem91@gmail.com

Alessandra Agostini
alessandra.agostini@disco.unimib.it

Rim Faiz
rim.faiz@ihec.rnu.tn

Gabriella Pasi
pasi@disco.unimib.it

[1] Department of Informatics, Systems, and Communication, University of Milano-Bicocca, Milan, Italy

[2] LARODEC, ISG, University of Tunis, Le Bardo, Tunisia

[3] Orange Developer Center, Orange, Tunisia

[4] LARODEC, IHEC, University of Carthage, Carthage Presidency, Tunisia