Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 333–340

Procedia Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 6 - 9, 2020, Warsaw, Poland

A Multi-Level Tourism Destination Recommender System

Hend Alrasheed*, Arwa Alzeer, Arwa Alhowimel, Nora shameri, Aisha Althyabi

Department of Information Technology, King Saud University, Riyadh, Saudi Arabia

Abstract

There are multiple factors that play a significant role in determining a tourist choice of a vacation destination such as affordability, availability of activities, popularity, and safety. Despite the mass of content available on the World Wide Web, the efficiency of utilizing it to find a destination that meets all the criteria of a potential traveler is always questionable. Therefore, travel and tourism software tools tend to incorporate a recommender system component to enhance the quality of the service they provide. In this paper, we propose a simple multi-level tourism recommender system framework to assist potential travelers find the destination that best matches their preferences and requirements. The system incorporates two recommendation procedures: providing the user with a set of destinations liked by similar users to allow constructing a list of potential destinations. Then the system ranks the destinations on this list based on the user preferences and constraints.

Keywords: Tourism recommender systems; destination ranking; Web scraping; user preferences and constraints.

1. Introduction

For many people, tourism provides an opportunity to perform leisure activities and explore new cultures. With the advancement of information and communication technology and the development of the living standards of societies, the demand for tourism have grown drastically.

There are multiple factors that play a role in determining a tourist choice of a vacation destination such as affordability, availability of activities, popularity, and safety [7]. Despite the mass of content available on the World Wide Web, the efficiency of utilizing it to find a destination that meets all the criteria of a potential traveler is always questionable. This is mainly due to the fact that the available information is not personalized and does not match the user inquiry. Therefore, travel and tourism software tools tend to incorporate a recommender system component to enhance the quality of the service they provide.

* Corresponding author.
E-mail address: halrasheed@ksu.edu.sa

10.1016/j.procs.2020.03.047

Multiple tourism recommendation and tour planning systems have been proposed [1, 5, 18, 11, 12, 15, 20, 17, 13, 21, 6, 9]. There is also a number of commercial destination recommendation tools [25, 26, 27]. Generally, tourism recommender systems are difficult to design because of the amount of available information as well as the complex and multi-layered nature of the desires and needs of tourists [2]. In this paper, we propose a simple multi-level tourism destination recommender system framework to assist potential travelers find the destination that best matches their preferences and requirements. In our framework, a destination is a city. The system incorporates two levels of recommendations as each user request undergoes two levels of recommendations. The first level involves providing the user with a set of destinations that matches her preferences (based on the preferences of similar users). The second level ranks the set of destinations based on the user preferences and constraints.

A registered user will be asked to provide a set of potential destinations as well as other information such as the travel dates and accommodation budget. Providing the set of potential destinations by the user is important to increase the system accuracy. The system then will assign a score to each destination. Finally, an ordered list of destinations will be returned to the user. To increase the recommendations accuracy, each user destination recommendation request goes through two steps. First, based on her preferences, the user will be given a set of destinations. This set contains destinations that are popular among other similar users. Second, the system ranks the selected destinations based on sets of user preference attributes and user constraint attributes with the use of data scraped from different Web portals to consider the dynamic context of each trip. For example, the destination's weather, accommodation prices, safety level of the destination, and traffic level of the destination during the selected dates. The system also permits users rating destinations they have visited and use this rating to implicitly update the user preferences.

2. Related work

According to their recommendation techniques, recommender systems can be classified as knowledge-based, content-based, and collaborative-based. Knowledge-based recommender systems recommend items based on the preferences and needs that the user provide. Content-based recommender systems recommend items based on the user's rating history. Collaborative-based recommender systems recommend items that are popular among other users with similar interests. Hybrid recommender systems aim at combining elements from two or more recommendation techniques to mitigate the weaknesses of each single one.

There are different types of tourism recommendation and tour planning systems [1, 5, 18, 11, 12, 15, 20, 17, 13, 21, 6, 9, 22]. Those systems provide plans based on user location or by explicitly asking the user to provide the destination she is planning to visit. Another type of tourism recommender systems provide destination recommendations [10, 24].

TripMatcher [3] uses a decision tree to build a knowledge base for each destination. The tree is built using two data source: content and ratings provided by domain experts and automatically generated ratings through text-mining of product descriptions. It also employs an incremental rating scheme to evaluate each trip based on the user response.

Leal et al. [10] proposed a system which suggests a destination to the user based on textual reviews and machine learning techniques. The system uses semantic content-based filtering to provide personalized recommendations based on Expedia crowd-sourced hotel textual reviews. In [24], the authors propose a destination recommender system that uses opinion mining to build the profiles of user preferences and item opinion reputations. They also employ collaborative filtering for destination rating prediction.

Sun et al. [19] use Flicker's public geotagged photos as well as other context information (such as weather) to extract users travel preferences and build their travel history. Then they use this profile to generate recommendations. In [8], the authors propose a hybrid recommendation model using the user ratings, reviews, and social data.

A number of commercial destination recommendation tools are also available [25, 26, 27]. Those tools work as a destination search engine. They assign and evaluate the destinations based on a set of predefined attributes. Then they suggest destinations by finding those ones that match the user preferences. Those tools only use information explicitly given by the user and do not exploit the user's history.

3. Preliminaries and Definitions

The destination selection problem can be defined as the "process of choosing one destination among a number of alternatives for the purpose of fulfilling the travel-related needs at hand" [4].

Typically, the process starts by listing a number of alternative destinations. Then information about each destination will be collected and the list of destinations will be ranked according to a specific set of user criteria. The destinations at the top of the list will be kept for serious consideration.

In this paper, we define each destination d_i as a tuple $(x_1, ..., x_n)$, where each x_j is an attribute that is used to describe destination d_i and that affects the satisfaction of the user towards that destination. For example, available attractions, weather, safety level, etc.

Throughout this work, we use the term *attraction types* (denoted by A) to refer to classes of places or activities that tourists usually target. For example, beaches, water parks, and swimming belong to the *water activities* attraction type. We assume that there are m attraction types $A = a_1, ..., a_m$. Each attraction type a_j is represented as a pair (l_j, c_j), where l_j is the label of attraction type a_j and c_j is the number of places or activities that belong to the attraction type a_j in destination d_i.

The *weather* w_i of destination d_i is defined as $w_i = (t_i, pr_i, h_i)$, where t_i is the average temperature over a specific period of time, pr_i is the average chance of precipitation over a specific period of time, and h_i is the average humidity over a specific period of time.

A user profile u is a tuple $(p_1, ..., p_r, c_1, ..., c_q)$ where each p_i is a user preference attribute and each c_i is a user constraint attribute. User preference attributes are the ones that do not change frequently such as preferred weather, while user constraint attributes are the ones that change frequently such as the travel dates and budget.

4. Proposed System

The proposed system includes two main recommendation processes. First, providing the user with an attractive set of destinations. Second, ranking the list of destinations provided by the user from the most to the least recommended.

Next we first provide a high level description of the main components of the system. Then we provide a detailed description of each system component.

4.1. System Description

The system is a hybrid tourism recommender system. It is collaborative-based because it maintains a list of group-based popular destinations based on the preferences of similar users. It is also knowledge-based because explicit user preferences are collected and matched against the characteristics found about each destination.

Upon registration, the user can (optionally) provide her preferences. User preference attributes include attributes that do not change very often. In the current framework, the system maintains two types of user preference attributes: favorite types of attractions and preferred weather conditions.

The system incorporates two levels of recommendation processes. The first recommendation process involves providing the user with a set of destinations that matches her preferences. To do so, the system selects those destinations that are popular among similar users. User similarity is based on the preference attributes of the system users.

The second recommendation process starts when the user asks the system to rank the selected set of destinations. The system will ask the user first to enter information about her travel (user constraint attributes). In the current framework, this set includes the travel dates, the accommodation budget, and whether or not the destination needs to be kids-friendly. The set of constraints can be easily extended to include other attributes that change frequently (with every trip). For example, flight duration and flight preferences. All preferences and constraint attributes are optional (except for the dates constraint attribute).

After the user has finished her selections, all user information will be combined to create the user profile which will be sent to the system planner. The planner will run multiple Web scrapers to collect data about each destination on the list. Then it will assign a utility score to each destination. Finally, the system will rank the destinations based on their scores and return them to the user. Figure 4.1 shows the main system steps.

To avoid the problem of having a static user profile, the user can explicitly update her user preferences. Moreover, the system will implicitly update the user preferences whenever the user provides a rating for a particular destination (assuming that the user finished her trip).

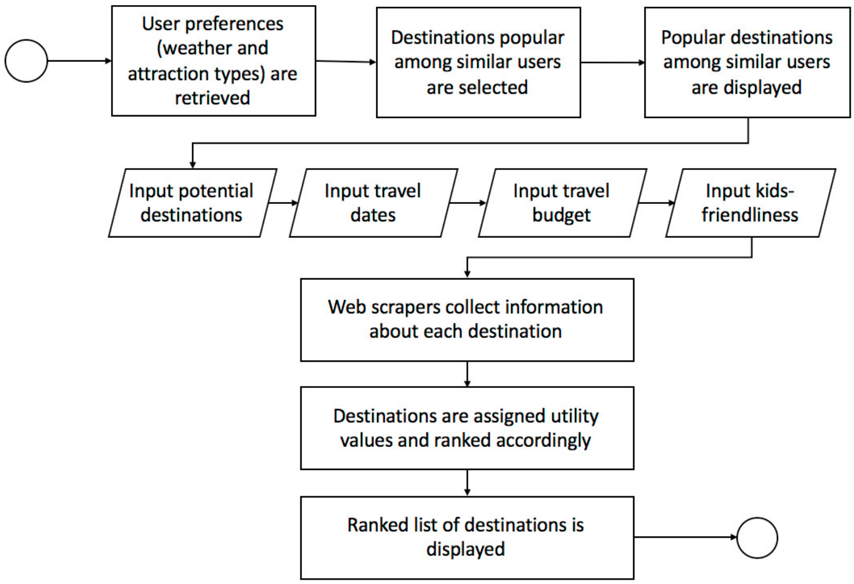

Fig. 1. Flowchart of the destination recommendation process.

4.2. System Details

The system architecture is presented in Figure 2. It has four main components: User profile, Group-based popular destinations, System planner, and Web scrapers. Next we provide a detailed description of each component. Then the user and system data will be covered.

4.2.1. User Profile

The user profile will be constructed each time a user requests a recommendation process. The user preference attributes (which have been collected during user registration) and the user constraint attributes (which have been collected when the user started the recommendation process) will be combined and structured according to a specific representation.

In the current framework, we consider two user preference attributes: weather and attraction types. We also consider the following user constraint attributes: travel dates, accommodation budget, and kids-friendliness. If a user does not provide a value for a particular attribute, it will not be included in the recommendation process. That is, a missing attribute will not be considered as one of the factors that impact the score of a destination. The only required attribute is the travel dates.

To elicit user preferences, the system uses an image-based approach. For example, to choose preferred weather, the system displays a set of images depicting different weather conditions. A user should identify those images that match her preferences. Each image is associated with a temperature range which will be stored as part of the user's preference attributes. Similarly, to choose preferred attraction types, the system displays a series of images, each of which represents a specific attraction type. The keyword associated with the attraction type will be saved to the user preferences.

4.2.2. Group-based Popular Destinations

The first level of recommendation in the system asks the user to choose a set of potential destinations. To simplify this process, the system suggests a set of destinations that are popular among similar users. That is, users who share

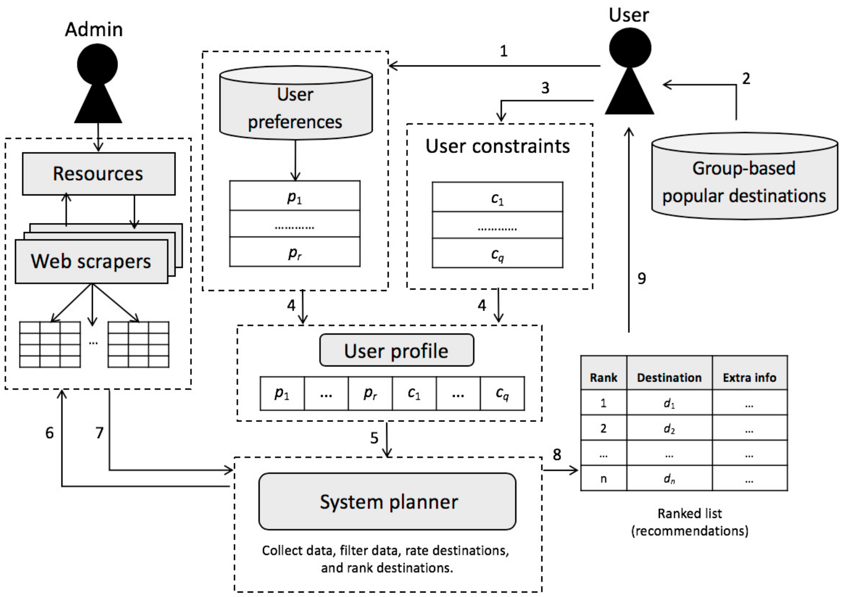

Fig. 2. System architecture. The numbered arrows show the data and activity flow between the main system components during the recommendation process.

similar preference attributes. This list of group-based popular destinations will be displayed to the user each time she starts a new destination ranking request. To measure popularity, the ratings provided by the system users will be used.

If the user chooses not to provide her preference attributes, the set of similar users cannot be identified. In this case, the system will provide a pre-defined list of possible destinations for the user to choose from.

4.2.3. System Planner

The System planner takes the user profile as input and returns a ranked list of destinations to the user. To achieve this goal, the system first needs to collect information about each destination using a set of Web scrapers. The Web scrapers data need to be structured in a way that matches the user profile representation. The system planner then performs a one to one comparison against the user profile attributes (Section 4.5).

4.2.4. Web Scrapers

The system planner uses a set of Web scrapers to collect data on each destination provided by the user. Each scraper retrieves data that is related to a specific user attribute (preferences and constraints). The scrapers performs text grepping to find the information. For comparison reasons, the user profile representation needs to match the destination representation.

Some of the information are available on the Web such as the predicted weather and the accommodation prices. Other pieces of information need to be calculated based on information found on the Web. For example, the safety level of a given destination which can be computed as an average value over the past few years.

4.3. User Data

To produce relevant and reliable recommendations, the system stores various pieces of data about each user. The user data comprises three main parts: the user demographic characteristics such as age, gender, and country of origin, the user preferences, and the user favorite destinations.

4.3.1. User preferences

An account with the user preferences is constructed by explicitly requesting different forms of preference information from the user. The goal is to keep a set of user preferences that is likely to remain unchanged over an extended period of time. The user will be able to explicitly update her preferences whenever she likes.

Moreover, the user preferences will be maintained by implicitly updating them each time the user provides a rating for a specific destination. This is expected to happen after the user had finished her trip. The system will store all the information related to the last destination ranking session and will use the data available for each destination to update the user preference attributes. For example, if a user did not include water activities as one of her preferred attraction types and provided a high rating after experiencing such attraction in the destination, then this attraction type will be added to her preferences. This new information will be used during the next destination ranking session. To avoid any undesirable change, the user will be notified and asked to confirm this update.

4.3.2. User destinations

Each time the user provides (or selects) a destination, it will be added to the user account. Note that some of those destinations will be rated by the user after visiting a destination. User destinations will be used to construct lists of group-based popular destinations and recommend them to similar users.

4.4. System Data

4.4.1. Group-based popular destinations

The process of constructing and maintaining the set of group-based popular destinations is as follows.

1. Each time a new user joins the system and provides her user preference attributes, a clustering algorithm will be applied. The clustering algorithm will identify groups of similar users. Moreover, the clustering algorithm will be applied whenever the preferences of a user are updated (explicitly or implicitly).
2. For each group, the system retrieves the destinations from all users and stores the average rating of each. Then the system sorts all destinations according to their new ratings and keeps only those destinations that exceeds a predefined rating threshold.
3. To display the list of group-based popular destinations to the user, the system retrieves all destinations assigned to the group to which the user belongs to. Then it cleans this list by removing some destinations including the place of residency and destinations with low ratings (if previously rated by the user).

4.4.2. System resources

The system resources includes all the Web portals used by the Web scrapers. For example, attractions data is available on https://www.tripadvisor.com, lodging data is available on https://www.booking.com, and weather data is available on https://www.holiday-weather.com.

4.5. Destination ranking

Let $\mathcal{D}$ be the set of potential destinations (provided by the user), $\mathcal{P}$ be the set of user preference attributes, and C be the set of user constraint attributes (user profile $u = \mathcal{P} + C$). The goal is to assign a score s_i to each destination $d_i \in \mathcal{D}$ based on the provided sets $\mathcal{P}$ and C. The scores will be used to rank the list of destinations and display them to the user from the most to the least suitable.

The process is as follows. For an active user u with a set $\mathcal{D}$ of potential destinations, the system assigns a utility to each destination d_i as a real valued function $R(u, d_i)$. The utility measures the degree to which the features of destination d_i match the attributes of user u. The value R needs to be assigned to all destinations, i.e., $R(u, d_1), R(u, d_2), ..., R(u, d_N)$, where N is the cardinality of set $\mathcal{D}$. The system then sorts all destinations in a non-increasing order (based on utility) and sends the list to user u.

The utility of each destination $d_i = (x_1, ..., x_n)$ is defined as follows.

$$R(u, d_i) = \sum_{j=1}^{n} v_j(x_j) \tag{1}$$

where n is the total number of attributes used to evaluate each destination and each v_j is the value function of attribute x_j. The higher the value of each attribute, the more it matches the user request.

In the current framework, we assume two user preference attributes: weather and attraction types and two user constraint attributes: accommodation budget and kids-friendliness ($n = 4$).

The value of each attribute function v_j is computed as follows.

1. The weather attribute denoted by v_1 is defined as $v_1 = \max(0, \min(e_u, e_i) - \max(s_u, s_i))$, where (s_u, e_u) is the preferred temperature range of user u and (s_i, e_i) is the predicted temperature of destination i based on the dates provided by the user.
   The average precipitation and humidity levels will be displayed to the user as extra information (see Section 4.6).
2. The attraction types attribute denoted by v_2 is defined as $v_2 = \frac{1}{\sum_{z=1}^{m} T - a_z(c_z)}$, where T is the total number of matching attractions and each a_z represents the number of attractions of type z that are available in destination d_i.
3. The accommodations attribute denoted by v_3 is defined as the percentage of accommodations that match the user's budget. An accommodation matches the budget of a user if its total cost is strictly smaller than or equal to the budget provided by the user.
4. The kids-friendliness attribute denoted by v_4 is defined as the percentage of attractions that are family-oriented.

4.6. Extra information

The system Web scrapers collect information about each destination. This information is necessary for the utility assignment and the ranking of the destinations. In addition, the Web scrapers collect information that can be displayed to the user such as the safety level and the traffic level in each destination. Those attributes do not factor in the ranking process, however they will be displayed to the user as extra information about each destination.

5. Final remarks

The publicly available information has a substantial influence on the choice of the travel destination. Tourism recommender systems can help users contend with information overload. This paper proposes a multi-level tourism recommender system framework to help users determine their most suitable destination. Each user destination recommendation request goes through two steps. First, based on her preferences, the user will be given a set of destinations. This set contains destinations that are popular among other similar users. Second, the system ranks the selected destinations based on sets of user preference attributes and user constraint attributes with the use of data scraped from different Web portals to consider the dynamic context of each trip

The current system uses the number of matching attractions and accommodations to decide the suitability of a destination. A future direction involves including the ratings of the attractions and accommodations to the decision of selecting a destination to enhance the accuracy of the recommendation process. Future work also involves the implementation of this system and the testing of its efficiency. There are several methods available to evaluate a proposed recommender system. Those methods can be partitioned into two classes: objective prediction accuracy and user subjective pinions. The results of the two mentioned classes of methods do not necessarily correlate [14].

To capture the users' subjective opinion, we will measure the users perceived qualities of our system using the ResQue framework proposed in [14]. We also plan to consider measure the system relevance by explicitly asking the user for her rating (a discrete numeric scale) after they visit a recommended destination. Finally, we will measure understandability and usability using the click-stream sequence analysis proposed in [23]. The click-stream sequence analysis measures the system understandability and usability by visualizing the users' interaction paths and detecting the number of users that choose to quit the recommender system.

Acknowledgment

The authors acknowledge funding from the Research and Development (R&D) Program (Research Pooling Initiative), Ministry of Education, Riyadh, Saudi Arabia.

References

[1] Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2003). Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices. Applied artificial intelligence, 17(8-9), 687-714.

[2] Berka, T., & Plnig, M. (2004). Designing recommender systems for tourism. Proceedings of ENTER 2004, 26-28.

[3] Delgado, J. A., & Davidson, R. (2002). Knowledge bases and user profiling in travel and hospitality recommender systems. na.

[4] Fesenmaier, D. R., Wber, K. W., & Werthner, H. (Eds.). (2006). Destination recommendation systems: Behavioral foundations and applications. Cabi.

[5] Garca-Crespo, A., Chamizo, J., Rivera, I., Mencke, M., Colomo-Palacios, R., & Gmez-Berbs, J. M. (2009). SPETA: Social pervasive e-Tourism advisor. Telematics and informatics, 26(3), 306-315.

[6] Gionis, A., Lappas, T., Pelechrinis, K., & Terzi, E. (2014, February). Customized tour recommendations in urban areas. In Proceedings of the 7th ACM international conference on Web search and data mining (pp. 313-322). ACM.

[7] Jensen, S., & Svendsen, G. T. (2017). What Determines the Choice of Tourist Destination? The Case of Denmark. safety, 5(2).

[8] Ji, Z., Pi, H., Wei, W., Xiong, B., Wozniak, M., & Damasevicius, R. (2019). Recommendation based on review texts and social communities: a hybrid model. IEEE Access, 7, 40416-40427.

[9] Kurata, Y., Shinagawa, Y., & Hara, T. (2015, September). CT-Planner5: a computer-aided tour planning service which profits both tourists and destinations. In Workshop on Tourism Recommender Systems, RecSys (Vol. 15, pp. 35-42).

[10] Leal, F., GonzlezVlez, H., Malheiro, B., & Burguillo, J. C. (2017, June). Semantic profiling and destination recommendation based on crowd-sourced tourist reviews. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 140-147). Springer, Cham.

[11] Luberg, A., Jrv, P., Schoefegger, K., & Tammet, T. (2011, September). Context-aware and multilingual information extraction for a tourist recommender system. In Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies (p. 13). ACM.

[12] Montejo-Rez, A., Perea-Ortega, J. M., Garca-Cumbreras, M. ., & Martnez-Santiago, F. (2011). OtiÅm: A web based planner for tourism and leisure. Expert Systems with Applications, 38(8), 10085-10093.

[13] Moreno, A., Valls, A., Isern, D., Marin, L., & Borrs, J. (2013). Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities. Engineering Applications of Artificial Intelligence, 26(1), 633-651.

[14] Pu, Pearl, Li Chen, and Rong Hu. ”A user-centric evaluation framework for recommender systems.” Proceedings of the fifth ACM conference on Recommender systems. ACM, 2011.

[15] Rey-Lpez, M., Barragns-Martnez, A. B., Peleteiro, A., Mikic-Fonte, F. A., & Burguillo, J. C. (2011, January). moreTourism: mobile recommendations for tourism. In 2011 IEEE international conference on consumer electronics (ICCE) (pp. 347-348). IEEE.

[16] Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). Springer, Boston, MA.

[17] Santiago, F. M., LPez, F. A., Montejo-REz, A., & LPez, A. U. (2012). GeOasis: A knowledge-based geo-referenced tourist assistant. Expert Systems with Applications, 39(14), 11737-11745.

[18] Sebastia, L., Garcia, I., Onaindia, E., & Guzman, C. (2009). e-Tourism: a tourist recommendation and planning application. International Journal on Artificial Intelligence Tools, 18(05), 717-738.

[19] Sun, X., Huang, Z., Peng, X., Chen, Y., & Liu, Y. (2019). Building a model-based personalised recommendation approach for tourist attractions from geotagged social media data. International Journal of Digital Earth, 12(6), 661-678.

[20] Vansteenwegen, P., Souffriau, W., Berghe, G. V., & Van Oudheusden, D. (2011). The city trip planner: an expert system for tourists. Expert Systems with Applications, 38(6), 6540-6546.

[21] Yang, W. S., & Hwang, S. Y. (2013). iTravel: A recommender system in mobile peer-to-peer environment. Journal of Systems and Software, 86(1), 12-20.

[22] Yu, Z., Xu, H., Yang, Z., & Guo, B. (2015). Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. IEEE Transactions on Human-Machine Systems, 46(1), 151-158.

[23] Zanker, Markus, et al. ”Evaluating recommender systems in tourisma case study from Austria.” Information and communication technologies in tourism 2008 (2008): 24-34.

[24] Zheng, X., Luo, Y., Xu, Z., Yu, Q., & Lu, L. (2016). Tourism Destination Recommender System for the Cold Start Problem. KSII Transactions on Internet & Information Systems, 10(7).

[25] http://www.jauntaroo.com.

[26] www.triporati.com.

[27] http://www.travelmyne.com