A Knowledge Graph for Travel Mode Recommendation and Critiquing

Bill Karakostas
VLTN GCV
Antwerp, Belgium
Bill.karakostas@vltn.be

Dimitris K. Kardaras
Athens University of Economics and Business
Athens, Greece
kardaras@aueb.gr

***Abstract*— The paper presents a knowledge based system for travel mode recommendation and critiquing. The system recommends the best travel mode for travelling between locations, based on user recommendations. The system's knowledge is stored in a graph database where the nodes represent locations and the edges the travel modes available for travelling between locations. Weights attached to each edge represent the degree of popularity of different modes for travelling on that route. The system is capable of recommending itineraries containing the highest recommended travel modes. The system also can critique a user proposed itinerary based on the travel modes it contains. We have evaluated the approach comparing system generated recommendations with user recommendations in online travel forums.**

Keywords- travel recommender system, graph database, multi mode travel, intelligent route planner, intelligent critiquing

I. INTRODUCTION

Research from the World Travel and Tourism Council (WTTC) indicates that the contribution of travel and tourism to world GDP grew for the sixth consecutive year in 2015, rising to a total of 9.8% of world GDP (US$7.2 trillion). According to WTTC, the tourism sector employs 284 million people, which globally represents 1 in 11 jobs. Stimulating demand and improving the traveler experience has been the endeavour of travel related commercial enterprises who are employing IT for that purpose. IT systems in various shapes (i.e., as static online information and advice, or through intelligent travel assistants) have been used to assist the travellers through the different stages of their trip, i.e., in planning, consuming, and also for post-travel feedback and ratings.

A particular class of intelligent information assistants known as *recommender systems* [1] has been used in the travel domain, to provide travellers with relevant recommendations regarding their trips. Some of the travel recommenders draw their knowledge from sources that describe travel and tourist locations, travel modes and other related aspects (called the *content based recommendation* approach), while others employ the experience of fellow travelers in order to provide relevant recommendations (an approach known as *collaborative filtering*). It has been suggested however, that most of the existing recommender systems only provide location-centric recommendations to travellers about 'things to do', once they get to their destination.

Some advanced recommenders, like SAMAP [2] and PaTac [3], are even capable of analysing the connection possibilities between the activities using different means of transport i.e., on foot, by bike, by car, or by public transport. This category of recommenders has similarities to automated travel planners. However, travel planners mainly rely on domain knowledge about routes and their properties, such as available travel modes, online timetables, knowledge of the average travel times and so on. Such knowledge is hard to acquire, integrate and maintain. On the other hand, travel knowledge elicited directly from the travel users themselves, maybe easier to acquire, due to the proliferation of travel related web sites such as forums. This knowledge may be less accurate and more subjective than the knowledge employed by travel planners, but that is compensated by the large volumes of available data.

Finally, another feature that travel planners are lacking is critiquing user proposed routes. Often users have a particular route in mind that they want to follow, but they want other user's opinion as to whether their route represents a good choice. A recommender system augmented with critiquing capabilities can comment on user proposals by comparing the users' routes (or the routes' legs, modes of transport etc), with what other users have recommended.

The paper therefore presents a travel route recommender and critique that does not rely on objective travel knowledge such as travel timetables, travel times and distances but on user recommendations. The system is capable of recommending the most popular means of transport between two locations. This differs from the typical travel planner's ability to find the best route between two places based on criteria such as travel time or cost. As argued above the route that optimises one or more of such parameters is not always the most popular with the users.

The structure of the paper is as follows. Section II surveys research approaches for travel and route recommendation. Section III presents the core of the approach including the architecture of the system, the organization of the knowledge base, the knowledge elicitation method and the implementation approach. Section IV describes the recommendation and critiquing algorithms, while Section V presents the testing and evaluation approach followed.

 ISBN: 978-1-61208-558-6

Finally, Section VI provides an appraisal of the significance of the work and its findings, as well as areas for future research.

II. INTELLIGENT SYSTEMS FOR TRAVEL AND ROUTE PLANNING AND RECOMMENDATION

Current literature shows that recommendation is a common service in the tourism subdomains of travel and travel services such as accommodation (i.e., hotel recommendations), used to make the site more appealing to users. Such recommender systems try to mimic the interactivity that occurs in traditional interactions with travel agents, for example when a user seeks advice on a possible holiday destination [4].

Some recommender systems recommend not only lists of places that match the user's preferences but also help to create a route through several attractions [5]. For example, CT-Planner [6] and [7] offers tour plans that can be refined gradually as the users express their preferences and characteristics (e.g., willingness to walk, walking speed, etc.). The recommender system described in [8] integrates automated selection of locations with finding the shortest path. Other recommender system takes into account factors such as the expected duration of the visit, the opening and closing times of the attractions and the distance between them. Examples of such systems include City Trip Planner [9], CRUZAR [10], Smart City [11], Otium [12] and e-Tourism [13]. Some advanced recommenders, are capable of analysing the connection possibilities between locations by different means of transport (walking, bike, car, public transport, etc.).

A related category of intelligent systems are Computer *Aided Critiquing systems*. The concept of a critique has been applied in diverse domains, including: medical, programming/software engineering and architectural design [14]. Critiquing system have been used by designers to improve their design artifacts by providing feedback. [15] analyse existing critiquing systems in terms of critiquing process, critiquing rules, and intervention techniques.

Travel planning however is a complex and dynamic process because there are multiple factors that influence the destination choice. Destination choice is determined by the availability of travel facilities and by the user's preferences such as length of travel, mode of transportation, accommodation type, and activity theme. Our approach exploits the benefits of itinerary planning using established graph search techniques to reduce the planning effort and the cost of information search for travellers.

III. SYSTEM ARCHITECTURE

Because planning is an inherently hard problem where optimal solutions often can only be approximated. Some planning systems for tourists reported are therefore not based on recommendations but instead use classic planning approaches like Operations Research techniques [8]. Our approach uses recommendations as heuristic planning rules that enable users to travel between the locations they want to visit in the most recommended way. Such recommendations act as shortcuts that reduce the space search effort in finding suitable routes between locations. Our approach therefore, combines the benefits of collaborative filtering with those of knowledge based approaches.

As shown in Figure 1, the knowledge of the travel mode recommender is captured in a knowledge graph linking locations with the most recommended modes of travel. The following section discusses the structure and content of the travel knowledge graph, while section IV presents the recommendation and critiquing processes.

The KB contains knowledge about tourist locations as well as of the recommended ways for travelling between locations, weighted by the degree of their recommendation. The knowledge is represented as a directed cyclic graph; we call the *travel knowledge graph*. Thus locations are represented as graph nodes and travel modes as graph vertices (edges). For example, for travelling between locations (nodes) A and B there can be *n possible travel modes* $m_1, m_2, \ldots, m_n$ where each mode m_k has a weight w_k (a real number greater than zero) that represents the degree of recommendation of that travel mode.

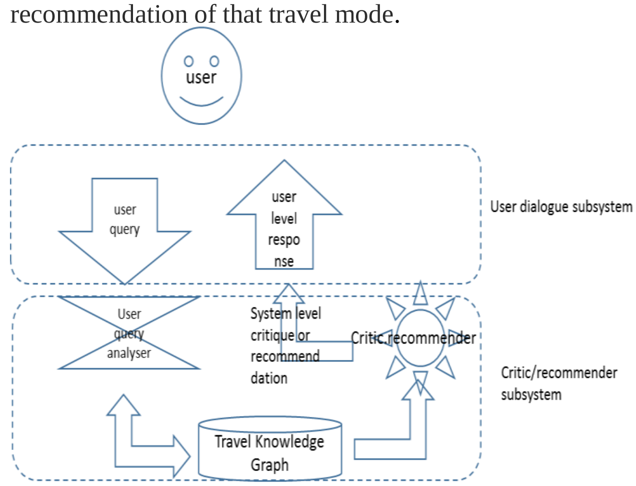

Figure. 1 Architecture of the recommender/critique system

When it is not possible to travel directly between two nodes A and B, a path P (called an itinerary in our approach) consists of intermediate nodes I1, I2,…, Ik connected by edges that form a path between A and B. Itineraries do not have to be acyclical graphs, i.e., a node can appear multiple times in an itinerary. Itineraries have to be finite however.

An optimal itinerary for travelling from A to B consists of a path I1, I2,…, Ik where every edge connecting two nodes In-1 and In corresponds to the highest recommended travel mode between the two locations.

In our knowledge graph model nodes represent cities, towns and other geographical areas of visiting interest or acting as hubs, i.e., facilitating the travel to other locations. Every node is connected to at least one other node via at least one edge that corresponds to a particular travel mode (road, rail, car, ferry, etc). Two nodes can be connected by multiple edges, representing the fact that locations are usually connected with multiple modes of transport. This approach represents an abstraction of a physical transportation network. For example, in a physical transportation network there may be a road linking A to B. In our approach however the links represent modes of transport, not physical connections, thus there will be two or more separate edges connecting A and B, representing private/hired car and (public) bus, running over the same (physical) road.

If a particular transport mode is not available between two locations, then it is simply not represented in the graph, while if it is available but there are no recommendations for it, (or against it) is assigned a recommendation weight of zero. In our approach, the direction of the travel is important, therefore all links are directed. For example, when travelling from A to B, bus might be the recommended travel mode, while in the reverse order it may not be recommended, due for example to the overcrowding of the returning bus. The directionality of the links is taken into account by the travel planner module in our approach.

The recommendation weight of each transport mode (edge) between two locations has a weight that represents its degree of recommendation. Recommendations are calculated according to the frequency with which a particular mode of transport is commented (in a positive or negative manner) in the travel forums. All recommendations are normalised within a scale 1 to 5 (i.e., from 'not recommended' to 'highly recommended').

For experimentation purposes we decided to populate the knowledge base will cover the geographical region of Italy known as Amalfi coast. This is a very popular touristic area of Italy attracting millions of tourists each year and attracts large online discussions on forums such as TripAdvisor. For example, the Italy forum of Trip Advisor contained more than 362,000 topics and 2million posts, as of 2016. Nodes for the transport network were selected by identifying the most frequently mentioned locations around Amalfi coast and by consulting constructed using online GIS sources such as OpenStreet Map and Google Maps. Recommendation weights for travel modes between Amalfi cost locations were elicited from general and specific travel advice of expert users who have travelled the route more than two times, as per the example below.

Recommendation Advice:

High speed trains between major cities run faster than any car: Venice, Bologna, Florence, Rome, Naples and Salerno are all linked by bullet trains. .... The big sights of Italy (Rome, Florence, Venice, Sorrento/Naples/Capri/Amalfi, and Cinque Terre) are inconvenient by car and easy by public transportation.

The recommendation weights are calculated as follows:

Assume as set of locations $L = \{l_i\}$ where i=1,…,n indicate locations, i.e. nodes on the travel graph that customers could potentially visit. Expert users' reviews are collected regarding the travel modes and their quality. Consider the set of travel modes $M = \{m_k\}$, where k=1,…,m the various available travel modes, such as private car, hired car, bus, train, plane, etc., for travelling between two locations. Let $E = \{E^q_{m_k,i,j}\}$ be the set of expert users' comments regarding the quality of m_k travel mode, of between two locations. Text mining tools such as the Knime can be used for analysing reviews and calculating the frequencies of terms related to travel modes and commented levels of travel quality. The E assumes five levels of travel quality (q) and uses linguistic variables and their corresponding fuzzy sets are shown below:

$$E = \begin{cases} verylow(0,0.10,0.25) \\ low(0.15,0.30,0.45) \\ medium(0.35,0.50,0.65) \\ high(0.55,0.70,0.85) \\ veryhigh(0.75,0.90,1) \end{cases}$$

Expert users comment on the suitability of a travel mode by using one of the above linguistic variables. Assume that $f^q_{m_k,i,j}$ indicates the frequency of using a linguistic variable $e^q_{m_k}$ to show the quality of travel by m_k travel mode, between two locations. By using the modal values of $e^q_{m_k}$ linguistic variables, the frequency of using a quality level is used to calculate the suitability of each travel mode as follows:

$s^{m_k}_{i,j} = \sum_{k,q} (f^{e^q}_{m_k,i.j} * e^q_{m_k,i,j})$, $\forall (i, j)$ location. Thus, $s^{m_k}_{i,j}$ shows the recommendation degree for travelling between locations by (i,j) by each m_k travel mode, i.e $s^{bus}_{i,j}$, $s^{privatecar}_{i,j}$, $s^{train}_{i,j}$, etc. The recommendation weight $r^{m_k}_{i,j}$ then for travelling between locations (i,j) is $r^{m_k}_{i,j} = \max\{ s^{m_k}_{i,j} \}$, with m_k indicating the most suitable thus, most recommended travel mode.

For an itinerary (I), all possible paths (P) on the travel graph that connect the departing (S) location and the destination (D) of a trip are considered. Thus, drawing on the cognitive maps theory, the recommendation weight for travelling between S and D is:

$R^{m_k}_{S,D} = \max\{ \prod_{i,j} (r^{m_k}_{i,j}) \}$, where S and D indicate the departing and destination locations respectively, the $r^{m_k}_{i,j}$ the recommendation degrees of each edge (i,j) along all possible paths, and the m_k shows the recommended travel mode.

Figure 2 shows a visual representation of the travel recommendation graph as implemented in the Neo4J graph database [16]. For visual clarity nodes of different type (e.g., city, town, village, see-sight area) are represented with different colour codes. Upon clicking on a node or edge the user can obtain information about the node attributes and their values. Neo4J has its own graph query language called

Cypher that was used to construct and run queries against the knowledge base.

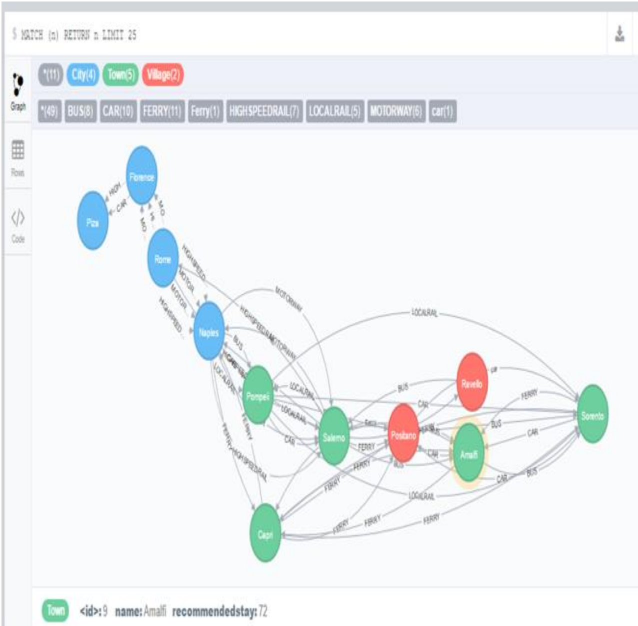

Figure. 2 Representation of the travel knowledge graph in the Neo4J graph database system

Figure 3 shows an example of such a query for constructing an itinerary for travelling between Rome and the town of Amalfi using only the highest recommended travel modes. Such plans are not optimal from a conventional planning perspective as they do not optimise travel time or distance, they represent however, the most popular ways that other travellers have used to travel between locations.

IV. THE RECOMMENDATION AND CRITIQUING ALGORITHMS

The recommendation algorithm can be formalised as follows: Given a user proposed tour consisting of locations l_1, l_2,…, l_k to be visited, recommend an itinerary that visits the required locations using the most recommended travel. The recommendation will consist of an itinerary l'_1 - m'_1 -> l'_2 m-'2 ->… , where l'_1, l'_2,… are locations from the user proposal with possible additional locations (hubs) added by the recommender system and each m'_k is the recommender way of travelling between locations l_k and l_{k+1}.

The following is a real user request from the TripAdvisor Italy Forum:

"What's the best itinerary for a 2 days trip to see Pompeii and the Amalfi coast from Rome…"

Producing a recommendation for the above requires the following steps:

Query analysis and constraint setting: Travel itinerary must include Pompeii. Paths between Rome and Amalfi will need to be produced. The query assumes returning back to Rome.

Query formulation in Cypher: This essentially involves formulating a query for finding all paths from Rome to Amalfi (that include Pompeii) and back, that use the highest recommended travel modes. The code snipper below shows the relevant query for finding all paths using Neo4J Cypher query language.

Cypher query: MATCH p=(a)-[*]->(b) WHERE (a.name = 'Rome') AND (b.name='Amalfi')RETURN DISTINCT nodes(p);

The above query returned 306 unique plans in the current version of the KB. Some of the itineraries are shown in figure 4. Heuristics can be used to prune the number of results by retaining paths that do not include too many revisits to the same locations and can fit within the time constraints set by the user (2 day).

```
MATCH (a)-[r*]->(b) WHERE (a.name = 'Rome') AND (b.name='Amalfi')
WITH  [x IN r WHERE x.recommended > 3] AS col
MATCH(a)-[x]->(b) WHERE x IN col RETURN DISTINCT  a.name,type(x),b.name
```

$ MATCH (a)-[r*]->(b) WHERE (a.name = 'Rome') AND (b.name='Amalfi') WITH [x IN r WHERE x.recommended > 3] AS col MATCH(a)-[x]->(b) WH…

a.name	type(x)	b.name
Rome	HIGHSPEEDRAIL	Naples
Naples	HIGHSPEEDRAIL	Salerno
Naples	LOCALRAIL	Pompeii
Pompeii	LOCALRAIL	Sorento
Sorento	FERRY	Amalfi

Figure. 3 Recommended travel modes between locations

1	"Naples"","Salerno"","Amalfi"
2	"Naples"","Salerno"","Naples"","Salerno"","Amalfi"
3	"Naples"","Salerno"","Naples"","Salerno"","Naples"","Salerno"","Amalfi"
4	"Naples"","Capri"","Naples"","Salerno"","Naples"","Salerno"","Amalfi"
5	"Naples"","Pompeii"","Salerno"","Naples"","Salerno"","Amalfi"
6	"Naples"","Salerno"","Pompeii"","Salerno"","Naples"","Salerno"","Amalfi"
7	"Naples"","Capri"","Positano"",,"Salerno"","Naples"","Salerno"","Amalfi"
8	"Naples"","Salerno"","Positano"",,"Salerno"","Naples"","Salerno"","Amalfi"
9	"Naples"","Pompeii"","Positano"",,"Salerno"","Naples"","Salerno"","Amalfi"
10	"Naples"","Capri"","Naples"","Salerno"","Amalfi"

Figure. 4. System produced itineraries

The critiquing algorithm can be formulated as follows. Given a user proposed route l1-m1->l2-m2>… compare the mode of each route leg with that which is highest recommended in the KB. Calculate an overall 'recommendability score' for the itinerary, for both cases, i.e. the one proposed by the user and the ones recommended by the system, by using algorithm and the formulas discussed in section III. The difference between critiquing and recommendation is that in critiquing the user itinerary is more detailed. The system does not propose a new itinerary but compares against the highest recommended one. This process can be iterative, i.e., the user can adopt her original plan, based on the received critique.

 ISBN: 978-1-61208-558-6

V. SYSTEM EVALUATION

Shani et al. [17] propose three different approaches for recommender system validation: offline validation, user studies and online experiments. In our approach because of resource constraints we opted for an offline experiment which however used real data both in terms in case of requests for recommendations and of actual recommendations taken from an online travel forum. While this approach is not as insightful as an online experiment, it can provide evidence of the performance of the recommender compared to actual users, without incurring the cost of user studies or online experiments. The objective of the evaluation was to test whether the recommender is exhibiting a behaviour that is close to that of the human recommender. Thus we had to find user recommendations for the same itinerary and compare the system produced recommendations to that of the average or typical user. We first however had to find a way to measure the similarity of recommendations. In our approach we opted for the overlap coefficient [18] (or, Szymkiewicz-Simpson coefficient) which is a similarity metric that measures the overlap between two sets, and is defined as the size of the intersection divided by the smaller of the size of the two sets, as shown in the following formula.

$$\mathrm{overlap}(X,Y)=|X \cap Y|/\min(|X|,|Y|)$$

We employ the overlap coefficient to two requests for recommendation cases described below. For each request we elicited user recommendations from the TripAdvisor Italy forum. These recommendations were not taken into account when populating the travel knowledge graph, hence they do not constitute 'training data' for the recommender. We construct the system recommendation using the approach described in Section IV and we compare it to each user recommendation to calculate an average overlap score between the system and the user recommendations. We also calculate the mean, variance and standard deviation of user recommendations overlappings to determine how much user recommendations overlap with each other.

TABLE I. CALCULATION OF OVERLAPPINGS FOR ROME TO POZITANO RECOMMENDATIONS.

Rec #id	User recom. avg. overlap	User recommendation	User-system Overlap score
1	0.5	Rome -high speed train -> Salerno- ferry-> Positano	1
2	0.5	Rome -high speed train -> Salerno- ferry-> Positano	1
3	0	Rome -high speed train -> Naples -car -> Sorrento -car-> Pompeii -car -> Sorrento -ferry -> Capri -ferry -> Sorrento -car -> Positano	0.43
4	0.375	Rome -high speed train-> Salerno -bus -> Amalfi -bus -> Positano	0.33
5	0.375	Rome -high speed train -> Salerno -taxi -> Positano	0.5
		Mean overlap of user recommendations: 0.35 Variance 0.0421 SD: 0.205	Average system-user overlap score:0.65

Table I shows what users actually recommended as itineraries for the Rome to Positano trip, how these recommendations overlap with each other on average and with the system recommendation. For this query, the system created the recommendation (Rome -high speed train -> Salerno- ferry->Positano) thus totally agreeing with the first two recommendations of Table I. Assuming a normal distribution in the overlapping values of user recommendations, we can observe that the system recommendation overlappings falls within two standard deviations of the mean, i.e., it has a typical overlapping (or similarity) to the user recommendations.

VI. CONCLUSIONS

Our approach integrates the formal/GIS view of travel planning (e.g., by following the shortest or the fastest route) with heuristic knowledge such as fellow user itinerary heuristics and recommendations that serve as shortcuts and help to reduce the cost of information search for the traveller. The attributes attached to nodes and edges can be extended with different features, reflecting other important travel considerations such as cost daily and seasonal variations. For example, roads that are very busy during the Summer period and thus get low recommendation might be more quiet in other seasons. Also, some modes along routes might be seasonal, for example some ferry lines might operate only in the Summer period.

The system could be extended with further reasoning capabilities, for example case based reasoning. Case based reasoning (CBR) has been already utilised in several recommender systems [19]. Previous travel experiences can be stored as cases in the knowledge base and new recommendations would entail recalling similar experiences from the knowledge base and reuse them partially, completely or modified.

Profiling could also be introduced to support more personalised recommendations. It has been argued however, [4] that in the case of travel this is very hard because each traveller's decision making profile is unique.

User proposed itineraries could be compared to existing recommended itineraries stored in the graph knowledge base. There is a lot of mathematical background in measuring graph similarity, for example by using distance measures like the Hamming distance, the simple matching coefficient, the Euclidean distance, and other metrics, and these could be utilised. However, such measures consider only little domain knowledge during the similarity assessment, while more sophisticated methods consider the different importance of individual attributes [20]. For example, two trips might visit the same locations [21] in the same order but the time spent on each location and the activities of the traveller could be very different.

ACKNOWLEDGMENT

Work described in this paper was partially funded by Horizon 2020 Research and Innovation Programme under Project 'Eutravel' (grant agreement No 636148).

REFERENCES

[1] J. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," In EC '99 Proceedings of the 1st ACM conference on Electronic Commerce, 1999.

[2] L. Castillo, et al., "Samap: An user-oriented adaptive system for planning tourist visits," Expert Systems with Applications, 34, 1318–1332, (2008).

[3] L. Ceccaroni, V., Codina, M., Palau, and M. Pous, "PaTac: Urban, ubiquitous, personalized services for citizens and tourists," In Third international conference on the digital society (ICDS 2009), February 1–7, 2009.

[4] F. Ricci, "Travel recommender systems," IEEE Intelligent Systems, 17(11/12), 55–5, 2002.

[5] P. Vansteemwegen, and W., Souffriau, "Trip planning functionalities:State-of-the-art and future," Information Technology and Tourism 12(4), 305-315, 2011.

[6] Y., Kurata, "CT-planner2: More flexible and interactive assistance for day tour planning. ENTER 2011. In Information and communication technologies in tourism (pp. 25–37), Innsbruck, Austria, 2011.

[7] Y., Kurata, and T., Hara. "CT-planner4: Toward a more user-friendly interactiveday-tour planner," In Information and communication technologies in tourism 2014 (pp. 73–86). Springer International Publishing, 2013.

[8] W., Souffriau and P., Vansteenwegen, "Tourist Trip Planning Functionalities: State–of–the–Art and Future Current Trends in Web Engineering," vol. 6385 of the series Lecture Notes in Computer Science pp 474-485, Springer 2010.

[9] P., Vansteenwegen, W., Souffriau, B., Vanden and D., Van Oudheusden, "The city trip planner: An expert system for tourists," Expert Systems with Applications, 38(6), 6540–6546, 2010.

[10] I., Mínguez, D., Berrueta, and L., Polo, "CRUZAR: An application of semantic matchmaking to e-tourism. In Y. Kalfoglou (Ed.), Cases on semantic interoperability for information systems integration: Practices and applications" (pp. 255–271). Hershey, PA: Information Science Reference, 2010.

[11] A., Luberg, T., Tammet, and P., Järv, "Smart city: A rule-based tourist recommendation system," In R. Law, M. Fuchs, & F. Ricci (Eds.), Information and communication technologies in tourism 2011. ENTER 11, January 26th–28th 2011, Innsbruck, Austria. Springer-Verlag, 2011.

[12] A., Montejo-Ráez, J., Perea-Ortega, M., García-Cumbreras, and F., Martínez-Santiago, "Otium: A web based planner for tourism and leisure," Expert Systems with Applications, 38, 10085–10093, 2011.

[13] L., Sebastià, I., Garcia, E., Onaindia, and C., Guzman, "E-tourism: A tourist recommendation and planning application," International Journal on Artificial Intelligence Tools, 18(5), 717–738, 2009.

[14] A., Mohd , J., Hosking, and J., Grundy, "A taxonomy of computer-supported critics," Information Technology (ITSim), International Symposium in, Volume: 3, 2010.

[15] O., Yeonjoo, M., Gross, and E., Yi-Luen Do, "Computer-aided critiquing systems. Lessons learned and new research directions. computer-aided critiquing systems," Lessons learned and new research directions, 2008.

[16] I., Robinson, J., Webber and E., Eifrem, "Graph Databases. New Opportunities for Connected Data," Second Edition O Reilly, 2015.

[17] G., Shani and Gunawardana, A. "Evaluating Recommendation Systems," Microsoft Technical Report 2009.

[18] D., Szymkiewicz and G., Simpson in http://paleo.cortland.edu/class/stats/documents/11_Similarity, last viewed December 2016.

[19] F., Ricci, D., Fesenmaier, N., Mirzadeh, H., Rumetshofer, E., Schaumlechner, A., Venturini, K., Wöber and A.H., Zins "Dietorecs: a case-based Travel advisory system. In Destination Recommendation Systems: Behavioural Foundations and Applications," (eds D.R. Fesenmaier, H. Werthner and K.W. Wöber) ©CAB International, 2006.

[20] A., Stahl, "Learning of knowledge-intensive similarity measures in case-based reasoning," PhD Thesis University of Kaiserslautern, 2004.

[21] A., Umanets, A., Ferreira, and N., Leite, "GuideMe – A Tourist Guide with a Recommender System and Social Interaction," Procedia Technology, Volume 17, pp. 407-414, 2014.