

Hotel Recommender System based on Knowledge Graph and Collaborative Approach

YOUSEF ABUZIR¹, MOHAMED DWIEB²

¹Faculty of Technology and Applied Sciences, Al-Quds Open University (QOU), Al-Madina Al-Munawara Street, Salfeet, Palestine
 (e-mail: yabuzir@qou.edu)

²Faculty of Technology and applied Sciences, Al-Quds Open University (QOU), The main road, Khallat Al-Badd, Beit Jala, Palestine
 (e-mail: mdweib@qou.edu)

Corresponding author: Yousef Abuzir (e-mail: yabuzir@qou.edu).

We gratefully thank Neo4j team for the great tool Neo4j, books and online resources of Neo4J that benefit us greatly in using neo4j in this research.

ABSTRACT With the rapid increase of Information technology, online services and social media, recommendation system becomes an important issue and a need for both the customer and business sectors. The main aim of traditional and online recommendation systems is to recommend the desired and the necessary services that are appropriate recommendations to users. Traditional recommendation systems often suffer from inefficient data analysis techniques, rating the different services without regard to the previous preferences of the users and do not meet the personal demands of the users. Therefore, in this paper we used a hybrid approach based on Knowledge graph and Machine Learning similarity function as a recommendation system. We used real datasets to conduct the experiment. We built the knowledge graph for the visitors, hotels and their ranks, and we used the knowledge graph and similarity scores to recommend a hotel or a set of hotels for the visitors based on former preferences and ratings of other visitors. The results show significant accuracy and good quality of service recommender systems with 93.5% for f-measure.

KEYWORDS recommender system, knowledge graph (KG), items based collaborative filtering, cosine similarity, function and prediction, Neo4j.

I. INTRODUCTION

RECOMMENDER systems [1] are support systems that help users in finding and selecting products (items) from given categories (i.e., accommodation, hotels, smart phones, movies, books, songs, insurance services, and so on) based on different and inhomogeneous information [2]. Schafer et al. [3], define recommender system as a tool designed to allow users to look through related knowledge that meets their interests and preferences.

Recommender systems are integrated in e-commerce applications and web applications to help the users make their choice. Hotels recommendation system is a hotel recommender system that helps users in selecting a hotel to book based on the preferences of users.

The main benefit of the recommender systems is providing better services for the customers. Using the information in the online hotel booking systems like scores or ranking, preferences and product evaluation by users, the companies are planning their goal based on the accuracy of the analysis of these data. The quality of the results may lead to more efficient planning, confident and better decision making and reduce cost and risk.

This paper discusses recommending a hotel to customers or visitors. In our approach, we use a database of hotel preferences and ratings based on customer stay. We collect all the hotel ratings the customer gave to a subset of hotels he visited.

Our recommender system calculates the similarities between two users based on hotel preferences and users

ranking for the hotels. Then the system generates the best hotels recommendation for the user. The similarity between users is calculated based on Cosine similarity function to recommend the best hotel for the customer.

Therefore, the need for an efficient and accurate recommendation system is very important and plays a crucial role in providing relevant information related to user or customer preferences, that is worthy of reliance or trust recommendations for the user.

A. KNOWLEDGE GRAPH

The Knowledge Graph is a model of a knowledge base created by Google in May 2012 and connects information in a visual way. It enhances the organic Google search engine's results. Google presents the information gathered from a variety of sources in an info box next to the search results.

One of the most common tools presenting knowledge graph or graph database is Neo4j. A graph database is used to represent relationships. An example of that is the Hotel Graph Database as well as the Recommendation relationships. You can see some of that in the graphic in Fig. 1. It is a sample graph Database from our hotel system using Neo4j. The circles represent nodes Visitors and Hotels. The lines, called edges, indicate relationships, the relation here is rated. Properties of that node like name of the hotel or the visitor are represented inside the circles.

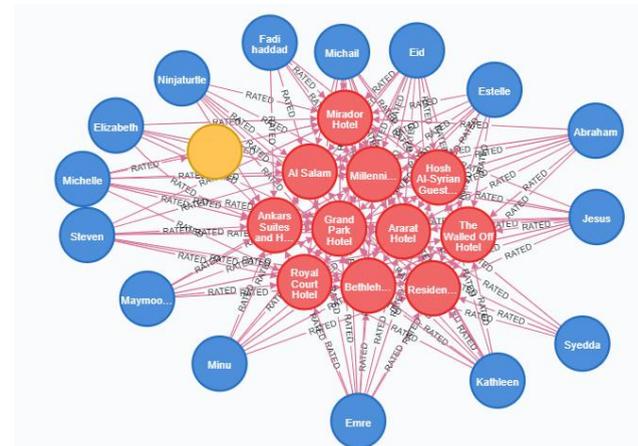


Figure 1. Sample of Hotel Graph Database

B. SIMILARITY MEASUREMENTS

There are different methods to calculate similarity between users and hotels using similarity measures (neighborhood techniques). Researchers use one of the following measures: Euclidian distance equation (1), Cosine distance equation (2) and Pearson correlation coefficient equation (3).

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}, \quad (1)$$

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}}, \quad (2)$$

where $\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ is the product of the two vectors

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

Cosine similarity measure is a metric used to determine how similar the items or visitors are. In our case, we are looking for hotels.

The rest of the paper is organized as follows. In Section II, we discuss related work on the different approach to recommendation systems and hotel recommendation. Section III, presents the proposed system based on knowledge graph technology was provided. Section IV discussed the implementation of the proposed system, experiment and results are discussed. Finally, Section V highlighted the conclusion.

II. RELATED WORK

There are several approaches to performing recommendation. Researchers used Collaborative, Content-based, Hybrid and Knowledge based approaches.

Collaborative approach [4, 5] is the most common used approach. This approach recommends Hotels or products by finding similar users to these hotels. It recommends hotels to you or the user based on their ranking and opinion [6-8]. Ringo is an online system that uses collaborative filtering (CF) approach and rating of the users on music albums to construct a user profile [9].

Content-based approach [10, 11], normally the system uses a user profile to give recommendation. This approach ignores all related information from other users [9, 12]. As an example, Letizia [13] is a recommendation system based on the behavior of the user on the user interface to advise him in browsing the Internet. The system tracks the user's browsing methods to predict web pages that are of interest to the user. Other researchers use the concepts of Neural Network (NN) in predicting and rating different Usenet news pages as either hot or cold [14]. Pazzani et al. [10] use the technology of intelligent agent and naive Bayesian classifier to recommend which web pages will interest a user.

Hybrid approach [15, 16] is widely used in recommendation system by combining two or more approaches. This approach is proposed to increase the quality, efficiency, accuracy and the performance of recommender systems [17, 18]. Cunningham et al. proposed a hybrid recommender system that combines two approaches the collaborative and content-based approaches [19]. Konstas et al. used a hybrid approach to design a music recommendation system, which combined tagging information, play counts and social relations [20]. Another work proposed a hybrid model that combines content-based with collaborative filtering (CF) for hotel recommendation. This model considers both hotel popularity in input destination and users preference. It produces the prediction with 53.6% accuracy [21]. Pawar et al. in [22] described their system KASR and the results showing that KASR significantly improves the accuracy and scalability of service recommender systems. They used techniques such as Map Reduce for parallel processing paradigm and Collaborative

approach for generating appropriate recommendations. Hassannia et. al. [23] purposed a hybrid recommendation system based on agent and web technologies.

Al-Ghossein, et al. [24] based on their approach learnt preferences for regions and used these preferences for hotel recommendation by mapping space of regions to user's preferences and computed the similarity which is called Recommendation System Based on Multi-Source Information [25].

The study of [26] proposed a recommender system that lists the name(s) of the travelers based on their preferences, by analyzing the other travelers' reviews together with the rating value to improve the prediction accuracy.

Finally, the knowledge-based approach is used to enhance the effectiveness and performance of the recommendation system using semantic classifications [27, 28]. This approach recommends items based on a predefined set of constraints (rules) and/or similarity metrics [29].

III. MATERIAL AND METHODS

A. BASIC MODELS OF RECOMMENDER SYSTEMS

There are different approaches to recommender systems and most of them are based on a single numerical value like rating to create recommendation services. In fact, evaluating

a service through only a single numerical value is not enough. However, evaluating service through multiple criteria and taking into account user references can help to make recommendations that are more effective for the users.

In our approach, we present a combined approach, which is based on preference of user and information related to hotels rating to implement a scalable recommender system by using Knowledge Graph and similarity functions techniques.

This section will show the steps used to build the graph database, add the customer data, hotels data and hotel ratings to our graph database and perform personalized recommendations

To evaluate the performances of the recommendation system, we select a list of users who visited mostly the same hotels dataset and combine their own ratings of these hotels. We then computed the similarity between the user ratings. Based on these measurements of similarities; the system suggests hotels to the customer to visit it.

In our approach, the system suggests hotel(s) similar to the ones visitor likes or suggests hotel(s) liked by other visitors who are similar to the customer. The system considers the rated of all the hotels that a user has given and then looks for hotels that are similar to what the user likes. Fig. 2 is a general architecture of our system.

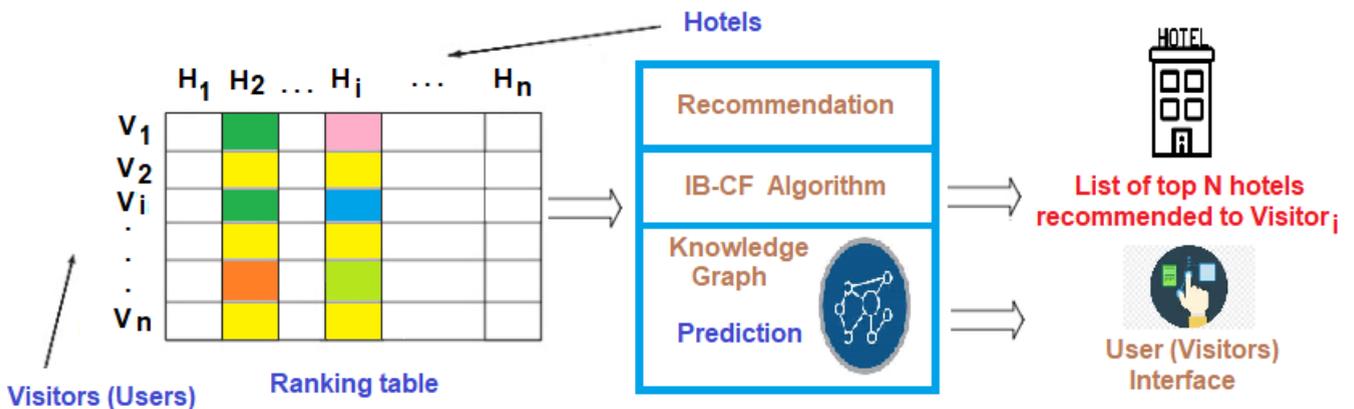


Figure 2. System Recommendation Architecture (Items Based Collaborative Filtering (IB-CF) and Knowledge Graph)

As seen in Fig. 2, Hotel-to-Hotel similarity is computed by looking into co-rating Hotels only. For example, for both Hotel H_2 and H_j , the similarity S_{ij} computed by looking into them. We have to remember that each of these co-rated pairs are obtained from different users. In Fig. 2, they come from visitor V_1 and V_i .

Referring to Fig. 2 as an example, each column can be viewed as a vector. Therefore, the item hotel H_1 has a vector and the item Hotel H_i has a vector of rated hotel. The similarity between these two vectors is measured by computing the cosine of the angle between them.

B. KNOWLEDGE GRAPH AND ITEM BASED COLLABORATIVE APPROACH

In our approach, we used Item-based Collaborative Filtering Algorithm. The approach computes how similar are the

target hotels i and then selects k most similar items based on the set of items the target visitors has rated. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target visitor's ratings on these similar items.

We implement the following strategy, which is based on visitor ratings. This means we can collect all the previous ratings of the visitors:

- To highlight subsets of visitors who are similar to one another, we collect all names of all hotels visited by the customers, what did they visit?
- Hotel Ratings, what ratings did they attribute?

To perform a recommendation for one given Visitor V_i , the ratings and recommendation of all its similar Visitors will be studied. To do that we followed the different steps (Fig. 3):

- Computing similarity – find the similarity between two Visitors V_i and V_j
- Visitors Selection – return Visitor who are the most similar to the Visitor you want to recommend hotels to, V_i
- Ranking hotels – define a way to rank hotels among the ones visited by the similar visitors
- Hotel Recommendation – recommend to V_i the hotels with best ranking, that he/she has not already visited.

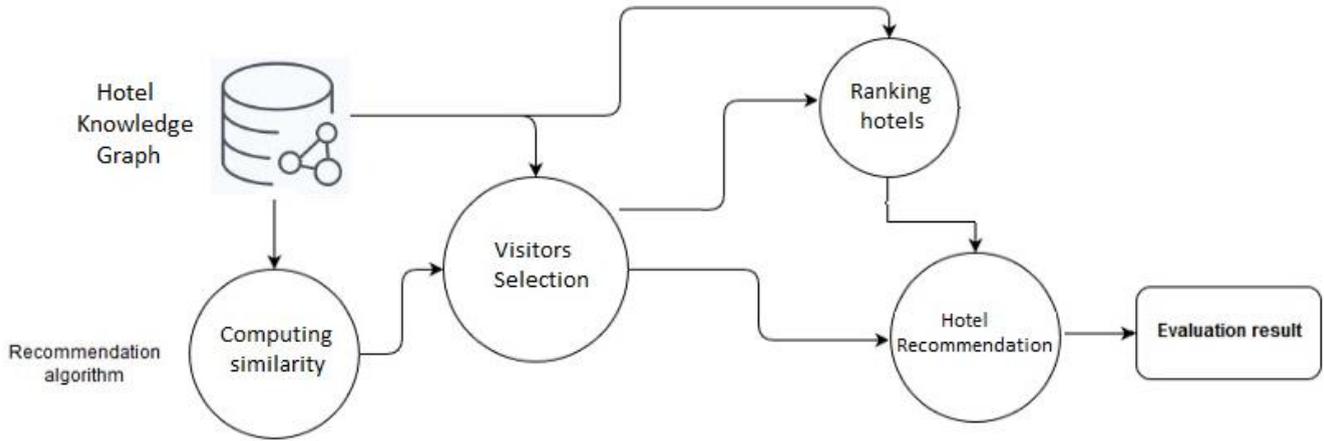


Figure 3. The recommendation flow

C. THE HOTELS DATASET

First, one has to build the graph database of hotels and visitors data that describe the dataset using Neo4j [30, 31]. We used the Hotels dataset we gathered on the Hotel Booking online website. This dataset consists of 125 ratings (4-10) from 15 users (visitors) in 11 hotels. The dataset has been cleaned up such that each user has rated at least more than one time. Data is gathered in the form of a set of excel file, the following paragraph describes dataset in Excel and in Knowledge Graph. As regards the Hotels dataset, it can be represented in the form of a very simple graph, with (Fig. 1):

- two kinds of nodes:
 - Person (visitor)
 - Hotels
- one edge:
 - Rated: If person V_i has rated hotel H_j , then V_i is explicitly linked with H_j , with a relation of type Rated. The rating $R(H_j, V_i)$ is recorded as a property of the edge.

D. INITIAL DATA MODEL

As we mentioned earlier our dataset consists of two types of nodes and one relationship. To add any new node or relationship to our graph database we use Create command. For example, to create a relationship called RATED we use the following command:

```
(Person)-[:RATED {rating:00}]->(Hotel)
```

Nodes- Based on the analysis of our data we gathered, we found that the database consists of Person (Visitor) nodes

and Hotel nodes. There are 15 persons and 12 Hotels in the dataset, where

1. Person – to create the first node for the person “Maymoona” we use:

```
CREATE
(Person1:Person {name:'Maymoona'}),
```

2. Hotel – to add hotel called “Ankars Suites and Hotel” to our databases, we used the following command:

```
CREATE
(Hotel1:Hotel {name:'Ankars Suites and Hotel'}),
```

Relation – the relation between nodes in our datasets is RATED. To create a relation between persons and hotels, where person1 rated hotel3 with a score of 6.7, and to add this relation to graph databases we used the create command too:

```
CREATE
(Person1)-[:RATED {rating:6.7}]->(Hotel1),
```

We added all the nodes and the relations to our databases. After creating our graph database we explored our graph database by using the following commands:

```
Match (n) Return n
```

Running the pervious command will show us the graphic representation of our databases. As shown in Fig. 4 we have 15 persons, 11 hotels and 125 rated relations.

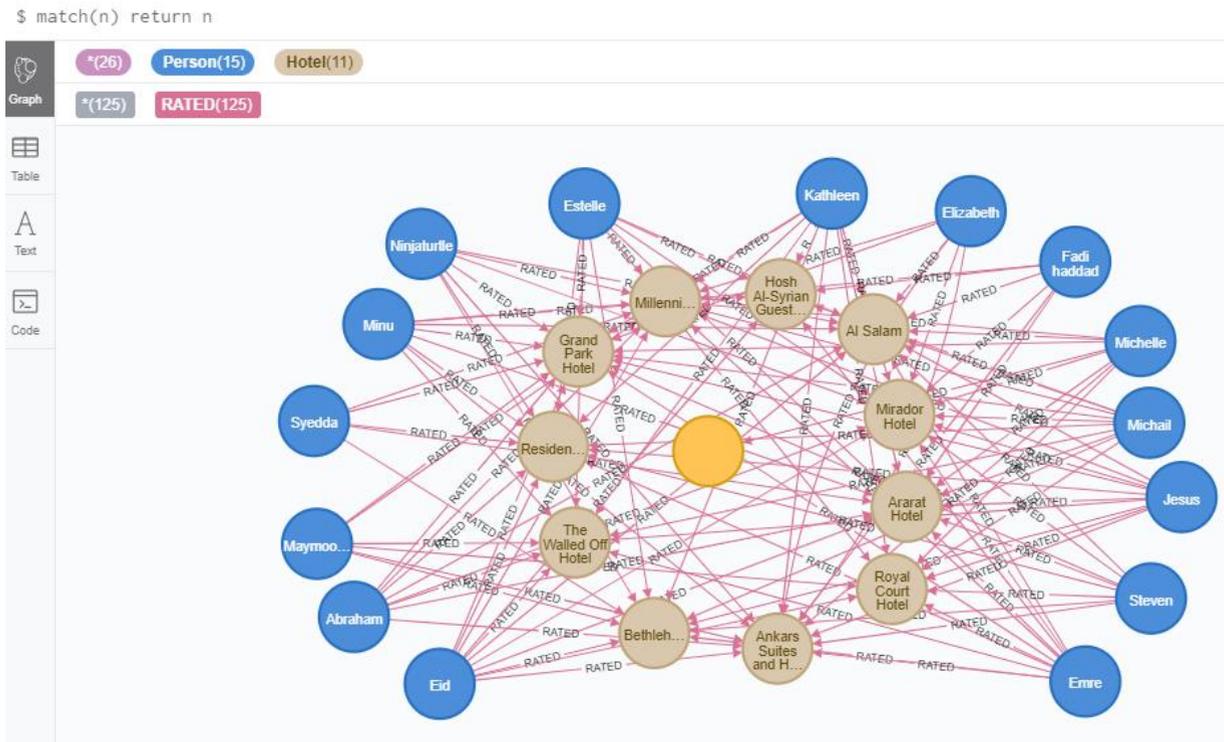


Figure 4. Graph Databases for our dataset related to Hotels and Persons

From Fig. 4, it is clear that the blue circles represent the persons and the gray circles represent the hotels, the line shows the relation Rated between the persons and the hotels.

IV. RESULTS AND DISCUSSIONS

In our approach we used the Item-Based Collaborative Filtering (IB-CF) to find the similarity. Therefore, the system recommends hotels to the visitors based on the similarity that the other visitor rated. To compute the similarity we can use a Cosine or Pearson Correlation Similarity. Referring to

Fig. 2, in IB-CF we based on vertical data not the horizontal data as in user-based Collaborative Filtering.

To compute the similarity, we used a [:SIMILARITY] [32] relationship between each person in the knowledge graph, the cosine similarity is considered as a property of the relationship. We used equation (2) to create the following Cypher query code to do that. Fig. 5 shows the knowledge graph after adding the similarity properties to the relation. The yellow lines represent the similarities.

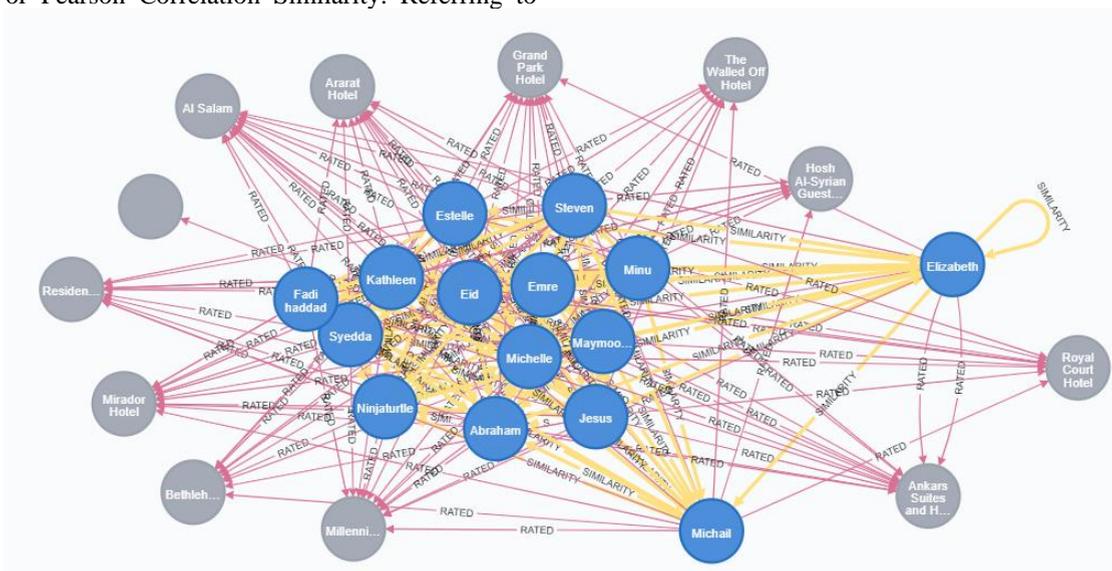


Figure 5. Hotel knowledge Graph with similarity relation

```
MATCH (p1:Person)-[a:RATED]->(H:Hotel)-[b:RATED]-(p2:Person)
WITH SUM(a.rating * b.rating) AS ProductTwoVectors,
SQRT(REDUCE(xDotProd = 0.0, x IN COLLECT(a.rating) |
xDotProd + x^2)) AS aDistance,

SQRT(REDUCE(yDotProd = 0.0, y IN COLLECT(b.rating) |
yDotProd + y^2)) AS bDistance,
p1, p2
MERGE (p1)-[s:SIMILARITY]-(p2)
SET s.similarity = ProductTwoVectors / (aDistance * bDistance)
```

If we are interested in computing the similarity between two visitors, we need to find the list of all hotels and the rating for both of them and then compute the similarity. For example, let us find the hotels that both 'Maymoona' and Minu rated. To do that we can write the following code to Cypher query our knowledge graph.

```
MATCH (p1:Person {name:'Maymoona'})-[r1:RATED]->(h:Hotel)-[r2:RATED]-(p2:Person {name:'Minu'})
RETURN h.name AS Hotel, r1.rating AS `Maymoona's Rating`, r2.rating AS `Minu's Rating`
```

The output of the Cypher query of the knowledge graph can be represented by graphical and tabular format. Fig. 5 shows the result in tabular format.

"Hotel"	" Maymoona's Rating"	"Minu's Rating"
"The Walled Off Hotel"	9	5
"Millennium"	8	7
"Grand Park Hotel"	7	8
"Ankars Suites and Hotel"	6.7	4.2

Figure 5. Cypher query result for hotels that both 'Maymoona' and 'Minu' rated

Now we can computed the similarity between the two visitors by querying our knowledge graph using cosine similarity code as follow:

```
MATCH (p1:Person {name:'Maymoona'})-[s:SIMILARITY]-(p2:Person {name:'Minu'})
RETURN s.similarity AS `Cosine Similarity`
```

The system computed the similarity between the two visitors as 0.96

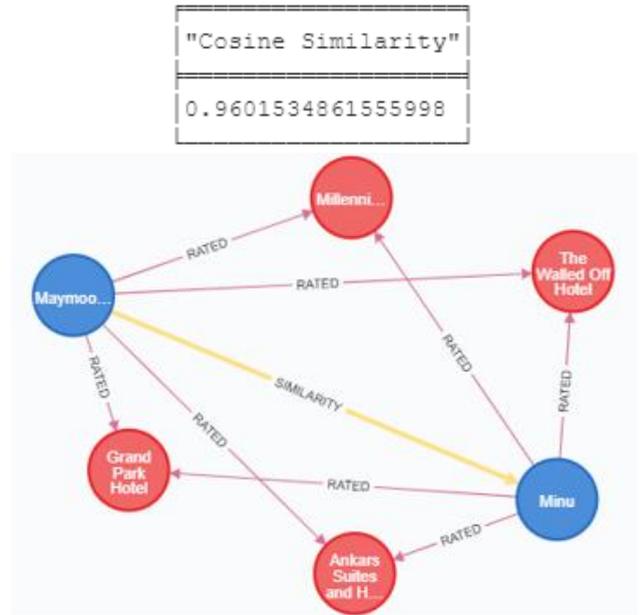


Figure 6. The similarity between the two visitors: Maymoona and Minu

We can view the k-nearest for example three k-nearest neighbors for the visitor 'Emre'. The higher the similarity the closer the neighbors.

```
MATCH (p1:Person {name:'Emre'})-[s:SIMILARITY]-(p2:Person)
WITH p2, s.similarity AS sim
ORDER BY sim DESC
LIMIT 3
RETURN p2.name AS Neighbor, sim AS Similarity
```

Fig. 7 presents the list of the visitors in descending order. The list shows the visitors who rated hotels most similarly to 'Emre'.

If we are interested in finding the list of the top three hotels, where they are rated based on similarity between visitors. We can create the following Cypher query, Fig. 8:

Neighbor	Similarity
"Michelle"	0.9969997101309398
"Jesus"	0.9889365283609625
"Elizabeth"	0.986371977387584

Figure 7. The list of the first three visitors who rated hotels most similarly to 'Emre'

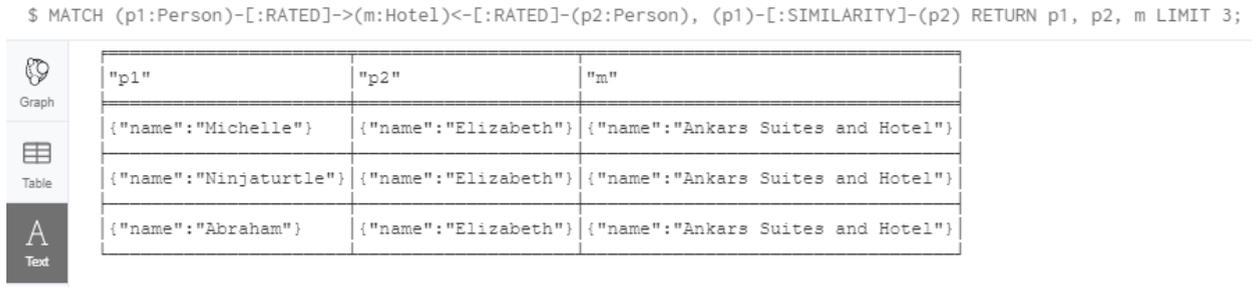


Figure 8. Top three hotels rated based on similarity

Another important function in our system is recommendation. A recommendation presented by recommending new hotels to the visitors that she/he did not book. To accomplish that we used Content-Based recommendation approach to implement the recommendation system that uses properties and users ratings of hotels. The fundamental assumptions from which recommendations began or developed or calculated are based on hotel recommendations to visitor V_i that are similar to previous hotels rated highly by other visitors V_j .

The following algorithm summaries the main steps:

1. Build hotel-profile in vector form where vector elements are features that can be binary (rating). In our knowledge graph, we have the nodes for visitors and hotels with preferences of booked and rated hotel.
2. Get all of the visitors who rated the hotels that Visitor V did not rate, their ratings for those hotels.
3. Calculate the similarity between all of the visitors who rated the hotels that Visitor V did not rate: Calculate cosine similarity equation (2).
4. Infer the recommendation by adding the vectors of the hotels rated by the visitors and compute the average vector (visitor-profile). By averaging the hotel ratings from that person's k -nearest neighbors out of the neighbors who rated the relevant hotel. Average the ratings in the ratings collection (equation (4)). The system returns the hotel name and the average rating as the recommendation:

$$recommend(h_j) = \frac{1}{(V_{sim} \cap Rh_j)} \sum_{v_i \in (V_{sim} \cap Rh_j)} rate(h_j, v_i), \tag{4}$$

where V_{sim} – the subset of visitors that have similar rating, Rh – the subset of visitors who have rated hotel h .

The following cypher code accomplishes the recommendation:

```
MATCH (b:Person)-[:RATED]->(h:Hotel), (b)-[:SIMILARITY]-
(a:Person {name:'Maymoona'})
WHERE NOT((a)-[:RATED]->(h))
WITH h, s.similarity AS similarity, r.rating AS rating
ORDER BY h.name, similarity DESC
WITH h.name AS hotel, COLLECT(rating)[0..3] AS ratings
WITH hotel, REDUCE(s = 0, i IN ratings | s + i)*1.0 / size(ratings) AS
recommend
ORDER BY recommend DESC
RETURN hotel AS Hotel, recommend AS Recommendation
```

Fig. 9 shows the list of recommended hotel with score in descending order of recommendation based on the similarity. The system returns the hotel name and the average rating as the recommendation to the visitor in the query “Maymoona”.

In this paper, we implement hotels recommender system based on similarity function. The principle is to use similarity cosine function to automatically identify other hotels that are similar to the ones the person likes.

Hotel	Recommendation
"Hosh Al-Syrian Guesthouse"	9.0
"Al Salam"	8.666666666666666
"Mirador Hotel"	8.666666666666666
"Residence Hotel"	7.333333333333333
"Ararat Hotel"	7.0

Figure 9. Recommending new hotels to the visitors that she/he did not book

The Hotel recommender system identifies Hotels that the user has rated and has not rated, and then suggests hotels very similar to his or her preferences and likes.

The performances of our hotel recommender systems are evaluated based on real cases. Based on our evaluation our hotels recommendation system that uses a hybrid knowledge based approach represented by the new technology of

knowledge graph and similarity measurement has proven to be quite effective for addressing that kind of issues.

We used three different scores to evaluate our recommendations system. These are Precision, recall and f-measure. Precision tells us how good our recommendation system is and can be calculated using the following equation $\text{precision} = (\text{relevant hotel} \cap \text{retrieved hotels}) / \text{retrieved hotels}$.

Recall tells the visitor that total amount of hotels retrieved are relevant, we can find recall using the following

formula: $\text{recall} = (\text{relevant hotel} \cap \text{retrieved hotels}) / \text{relevant hotels}$. We use the following formula to find f-measure = $(2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$.

To do that, we removed part of our dataset and asked our system to recommend the visitor a set of hotels. We selected the top five visitors who have visited similar hotels. We compared the new recommendations with the actual data, and then we found the precision, recall and f-measure. Fig. 10 represents the finding and the results.

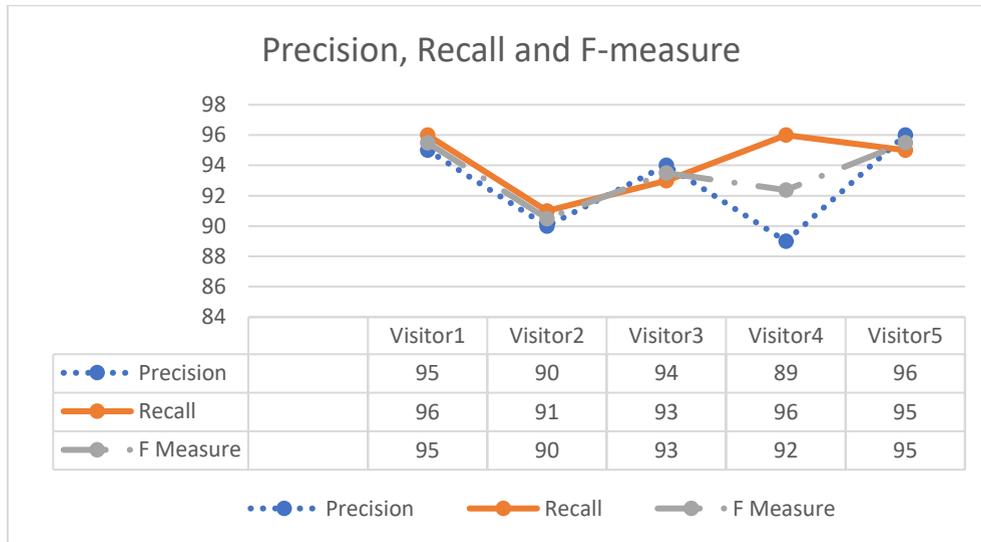


Figure 10 – System Evaluation using precision, recall and f-measure

The test of results showed that a good recommendation quality has been achieved with 92.8% Precision, 94.5% for recall and 93.5 for F-measure.

The results obtained from this experiment demonstrate the validity of this approach, by using the concept of knowledge graph and similarity function techniques to recommend a visitor relevant and interested hotels. Using such an innovative approach, we can achieve effectiveness, efficiency, interactivity, and high quality.

V. CONCLUSIONS

In this paper we proposed new two approaches: the similarity based approach and the knowledge graph approach to incorporate semantic and leveraging multi criteria rating information in recommender systems. Our recommender system will help users in choosing an appropriate option for accommodation.

We tested our approaches with a real dataset collected from websites for booking accommodation online. Based on the analysis of our datasets we evaluated our approach with respect to the degree of similarity between the visitors. The results of our evaluation show that our approach is able to identify recommendations of similar solutions in a highly efficient manner with 93.5% for f-measure. Further results and confirmations may be reached by extending the knowledge graph of datasets.

References

- [1] S. Bhatnagar, "Recommender systems in R," in *book Data Mining Applications with R*, 2014, pp. 117–151. <https://doi.org/10.1016/B978-0-12-411511-8.00005-0>.
- [2] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee and J. A. Konstan et al., "Getting to know you: learning new user preferences in recommender systems," *Proceedings of the International Conference on Intelligent User Interfaces*, 2002, pp. 127–34. <https://doi.org/10.1145/502716.502737>.
- [3] J. B. Schafer, J. Konstan and J. Riedl, "Recommender system in e-commerce," *Proceedings of the 1st ACM Conference on Electronic Commerce*, 1999, pp. 158–66. <https://doi.org/10.1145/336992.337035>.
- [4] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon and J. Riedl, "Applying collaborative filtering to usenet news," *Communication ACM*, vol. 40, issue 3, pp. 77–87, 1997. <https://doi.org/10.1145/245108.245126>.
- [5] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Model User-Adapt Interact*, no. 22, pp. 101–123, 2012. <https://doi.org/10.1007/s11257-011-9112-x>.
- [6] A. M. Acilar and A. Arslan, "A collaborative filtering method based on artificial immune network," *Expert Systems with Applications*, vol. 36, issue 4, pp. 8324–8332, 2009. <https://doi.org/10.1016/j.eswa.2008.10.029>.
- [7] L. S. Chen, F. H. Hsu, M. C. Chen and Y. C. Hsu, "Developing recommender systems with the consideration of product profitability for sellers," *Int J Information Sciences*, vol. 178, issue 4, pp. 1032–1048, 2008. <https://doi.org/10.1016/j.ins.2007.09.027>.
- [8] M. Jalali, N. Mustapha, M. Sulaiman and A. Mamay, "WEBPUM: a web-based recommendation system to predict user future movement," *Expert Systems with Applications*, vol. 37, issue 9, pp. 6201–6212, 2010. <https://doi.org/10.1016/j.eswa.2010.02.105>.

- [9] L. S. Chen, F. H. Hsu, M. C. Chen and Y. C. Hsu, "Developing recommender systems with the consideration of product profitability for sellers," *Int J Information Sciences*, vol. 178, issue 4, pp. 1032-1048, 2008. <https://doi.org/10.1016/j.ins.2007.09.027>.
- [10] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5(6), pp. 393-408, 1999. <https://doi.org/10.1023/A:1006544522159>.
- [11] D. Billsus and M.J. Pazzani, "A hybrid user model for news story classification," Kay J, editor, *Proceedings of the Seventh International Conference on User Modeling*, Banff, Canada. Springer-Verlag, New York, 1999, pp. 99-108. https://doi.org/10.1007/978-3-7091-2490-1_10.
- [12] S. H. Min and I. Han, "Detection of the customer time-variant pattern for improving recommender system," *Expert Systems with Applications*, vol. 37, issue 4, pp. 2911-2922, 2010.
- [13] H. Lieberman, "Letizia: an agent that assists web browsing," *Proceedings of the 1995 International Joint Conference on Artificial Intelligence IJCAI*, Montreal, Canada, 1995, pp. 924-929.
- [14] A. Jennings and H. Higuchi, "A personal news service based on a user model neural network," *IEICE Trans Information Systems*, E75-D(2), pp. 198-209, 1992.
- [15] R. Burke, "Hybrid web recommender systems," In: Brusilovsky P., Kobsa A., Nejdil W., editors. *The Adaptive Web, LNCS 4321*. Berlin Heidelberg, Germany: Springer, 2007, pp. 377-408. https://doi.org/10.1007/978-3-540-72079-9_12.
- [16] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, *Recommender Systems – An Introduction*, Cambridge University Press, 2010. <https://doi.org/10.1017/CBO9780511763113>.
- [17] B. Mobasher, "Recommender systems," *Kunstliche Intelligenz, Special Issue on Web Mining*, Bottcher IT Verlag, Bremen, Germany, vol. 3, pp. 41-43, 2007.
- [18] G. Murat, G.O. Sule, "Combination of web page recommender systems," *Expert Systems with Applications*, vol. 37, issue 4, pp. 2911-2922, 2010. <https://doi.org/10.1016/j.eswa.2009.09.046>.
- [19] P. Cunningham, R. Bergmann, S. Schmitt, R. Traphoner, S. Breen and B. Smyth, "WebSell: intelligent sales assistants for the world wide web," *Proceedings of the CBR in ECommerce*, Vancouver BC, 2001, pp. 104-109.
- [20] I. Konstan, V. Stathopoulos and J. M. Jose, "On social networks and collaborative recommendation," *Proceedings of the 32nd International ACM Conference (SIGIR '09)*, 2009, pp. 195-202. <https://doi.org/10.1145/1571941.1571977>.
- [21] J. Wang, J. Sun and Z. Lin, "Hotel recommendation based on hybrid model," 2016. cs229.stanford.edu/proj2016spr/report/041.
- [22] J. S. Pawar, N. R. Patil, H. R. Khide, S.S. Dandge and S. S. Bhavsar, "Recommender system for hotels based on user's personalized ratings using hadoop and cloud," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, issue 2, pp. 97-102, 2016.
- [23] R. Hassannia, A.V. Barenji, Z. Li and H. Alipour, "Web-based recommendation system for smart tourism: Multiagent technology," *Sustainability*, no. 11, paper 323, 2019. <https://doi.org/10.3390/su11020323>.
- [24] M. Al-Ghossein, T. Abdessalem and A. Barre, "Exploiting contextual and external data for hotel recommendation," *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP'18)*, 2018, pp. 323-328. <https://doi.org/10.1145/3213586.3225245>.
- [25] Y. Guo, C. Yin, M. Li and P. Liu, "Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business," *Sustainability*, vol. 10, no. 1, paper 147, 2018. <https://doi.org/10.3390/su10010147>.
- [26] U. Fasahte, D. Gambhir, M. Merulingkar, A. Monde and A. Pokhare, "Hotel recommendation system," *Imperial Journal of Interdisciplinary Research (IJIR)*, vol. 3, issue 11, pp. 318-324, 2017.
- [27] M. Atas, N. T. T. Tran, A. Felfernig, S. P. Erdeniz, S. Ralph and S. Martin, "Towards Similarity-Aware Constraint-Based Recommendation," in book: *Advances and Trends in Artificial Intelligence. From Theory to Practice*, 2019. https://doi.org/10.1007/978-3-030-22999-3_26.
- [28] A. C. Aggarwal, *Recommender Systems: Technologies and research issues*, "Proceedings of the 10th Int. Conf. on Electronic Commerce (ICEC'08)", Innsbruck, Austria, 2008, article no. 3, <https://doi.org/10.1145/1409540.1409544>.
- [29] C. C. Aggarwal, *Recommender Systems: The Textbook*, 1st ed., Springer, 2016. https://doi.org/10.1007/978-3-319-29659-3_1.
- [30] R. V. Bruggen, *Learning Neo4j*, PACKT Publisher, Birmingham, Mumbai, 2014.
- [31] I. Robinson, J. Webber and E. Eifrem, *Graph Databases*, 2nd ed., O'Reilly Media, 2015.
- [32] M. Needham and A. E. Hodler, *Graph Algorithms, Practical Examples in Apache Spark and Neo4j*, O'Reilly Media, 2019.



Prof. YOUSEF ABUZIR holds a Ph.D. in Computer Engineering and Applied Sciences from Ghent University in Belgium (2002). He is the Editor in Chief of the *Palestinian Journal of Technology and Applied Sciences (PJTAS)*. He served as a coordinator for different Tempus and Erasmus+ coordinator (2005-current) and ICM Erasmus+ Coordinator at QOU. Abuzir's research has resulted in four books and more than 70 journal articles, conference and book contributions in the fields of Artificial Intelligence, Information Retrieval, Knowledge Discovery, e-Learning, Natural Language Processing, Data Mining, Bid Data and Cloud Computing.



Dr. MOHAMED DWEIB, is currently a member of CIS Department in Technology and Applied Sciences program at QOU. He holds PhD degree in CS from Wroclaw University of Technology, Poland, 1999. He worked as a lecturer in QOU in the field of CIS, Bethlehem- Palestine.

Areas of scientific interests are: Information Retrieval and Expert system, Natural Language Processing and e-learning.

...