Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 8, 2020

GRAPH-POWERED RECOMMENDATION ENGINE IN MOVIE RECOMMENDER SYSTEM

Ratna Dewi Abdul Mutalib[a], Saadah Hassan[*a], Chew Yew Chong[b], Novia Admodisastro[a], Salmi Baharom[a]

[a]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia
[b]RHB Bank Berhad, Malaysia
Email: saadah@upm.edu.my

Received: 20.05.2020 ***Revised: 17.06.2020*** ***Accepted: 06.07.2020***

Abstract
Usually, people will search on the Internet for movie that they want to watch. However, it is tedious to find and choose movie that matched with their preferences due to a lot of information on the Internet. Therefore, most of the movie portals are adopting recommendation engine to filter and display user's personalized content. In this paper, MyRecMovie is presented to recommend movies by using graph-powered recommendation engine. MyRecMovie adopts content based (CB) and collaborative filtering (CF) approaches with then further enhance the recommendations with graph-powered recommendation engine to provide movie recommendations to the user.

Keywords--content-based filtering, collaborative filtering, graph database, recommender systems

DOI: http://dx.doi.org/10.31838/jcr.07.08.261

INTRODUCTION

The common question when we decided to watch a movie is 'what to watch today?' Then, we will start looking from the Internet about movies that we are interested to watch. We can do search by movie's name, actors, directors, or genre. Though, it is tedious and takes some time in retrieving relevant informationdue to information overload (Shah Khusro, Zafar Ali & Irfan Ullah, 2016). Therefore, many movie portals implement recommendation system (e.g., Netflix, YouTube, iflix) to filter and predict movies that users might like to watch. It reduces the user's effort to search and easier to make decisions. Besides, many e-commerce and website implement the recommender system to boost their profit and increase their sales. For example, Amazon, where 35% of purchases are results of the recommender system implemented on its site (MacKenzie, Meyer, & Noble, 2013). While, more than 70% of the time people spend on YouTube is because of the recommender system (Solsman, 2018) and Netflix can save around $1 billion each year after employing a recommender system (Gomez-Uribe & Hunt, 2016).

There is a large amount of information on the Internet that need to be filtered in order to make it closer to user's preferences. The common recommendation approaches implemented in the recommender system are content-based (CB) filtering and collaborative filtering (CF). CB filtering approach is based on characteristic information of an item (e.g. reviews, genre, and categories) and users (e.g., profile, demographic, interest). The recommender system will recommend items similar to what the users havepreviously searched, viewed or bought. While the CF approach is focusing on user-item interactions like rating or reviews, recent purchase, and recently like by other users. The recommendations are based on users with similar rating behavior that means if they show a lot of interest on a particular item, the recommendation system will recommend that item.

This paper presents a movie recommender system called MyRecMovie. The collaborative filtering (CF) approach and content-based (CB) filtering approach are used to predict what the users might like to watch. CB approach is where the system recommends the same movie genre with the movie that has been reviewed by the user. While, CF approach is where the system recommend movies to users based on their profile and other users' profiles. Furthermore, graph database is used in MyRecMovie as it was mentioned that Neo4j query results are much faster than MySQL (Vukotic, A., *et al.*, 2014).

The remaining of this paper is organized as follows: Section 2 discusses the related work, Section 3 explains the development of MyRecMovie, and Section 4 discusses the results. Finally, Section 5 concludes this paper.

RELATED WORK

Recommender system is proposed to tackle issue regarding information overload (Konstan, J.A., &Riedl, J., 2012). For example, recommender systems able to help e-commerce industries to enhance their strategies in selling their products (Hu, R., &Pu, P., 2009; Hussain et al., 2016; Pu, P., Chen, L., &Hu, R., 2011). In addition, recommender systems have also proved to improve decision making process and quality (Pathak, B. *et al.*, 2010). In which, it help users to find and select products in an online shopping environment (Hu, R., &Pu, P., 2009). Recommender system has the ability to predict the users' preferences and interests by studying their behavior (Resnick, P., & Varian, H. R., 1997). Wherein, it is used to filter a large amount of data and generate the most relevant information that the users' might interested to know (Pan, C., &Li, W., 2010). They can recommend the most suitable products or services to the intended users based on the information collected either by explicitly (usually by collecting users' ratings) or implicitly (usually by monitoring users' behavior such as product bought) (Lu, J. *et al.*, 2015; Bobadilla, J., 2013). Generally, recommender systems are beneficial to both service providers and users (Pu, P., Chen, L., &Hu, R., 2011).

There are two approaches commonly used to implement recommender systems, which are collaborative filtering (CF) and content-based filtering (CB). CF is an approach that helps users to choose from a range of recommendations based on the opinions of other users who share the same interests (Lu, J. *et al.*, 2015). For example, the CF approach provides recommendations by identifying the similarities among users' profiles and product's ratings. The CF approach can be divided into two types which are the user-based and item-based type. For user-based

type, the user will receive recommendations of items or products liked by similar users. For the item-based type, the user will receive recommendations of items or products that are similar to the items or products that they have loved back then. While, the CB filtering approach work by comparing representations of details of an item that the user has liked or reviewed in the past (Melville, Mooney, & Nagarajan, 2002). Unlike the CF approach, CB's recommendations normally based on user's information, and ignore the contributions from other users (Min, S.H., &Han, I., 2010; Hussain et al., 2016; Celma O., &Serra, X., 2008). In general, CB filtering approach relies on the similarities of the products and users such as genres, type, profile and categories. While, CF approach relies on how other users responding to the same products (user-item) such as their rating, liked, purchased and watched. Realizing the benefits, there are various applications that have utilized the approaches (e.g., Acilar A.M., & Arslan, A., 2009; Chen, L.S., Hsu, F.H., Chen, M.C., & Hsu, Y.C., 2008; Jalali, M., Mustapha, N., Sulaiman, M., & Mamay, A., 2010).

Graphs are natural mathematical structures allowing encoding possible interactions between different entities; recommendations can be computed through graph traversals (Joseph P., 2017). A graph database models and stores data as *nodes* and *edges* of a graph structure (Angles, Renzo & Gutierrez, Claudio, 2008). Nodes can represent anything such as person and product, while edges will connect the nodes. Figure 1 shows an example of the graph database model where the objects represented by numbers will be the nodes and each node will be connected to each other by the arrow when they have a relationship in sharing their attributes. That is the concept of graph that differs from a common database that uses a primary key to connect between data and it will take up more spaces and times to trace their similar behavior. But, using the graph, we can see clearly what and who is connected to each other even though there is no primary key has been set. The significant advantage of graph database is it allows efficient and fast retrieval of complex hierarchical structures that are difficult to model in relational systems (Joseph P., 2017). Moreover, information retrieval of graph database is performed based on the semantic of graph traversals and the size of graph database, thus, it does not influence the traversal performance(Joseph P., 2017). Therefore, several kinds of recommendations for a given dataset can be elaborated by combining and scoring paths between entities (Joseph P., 2017).

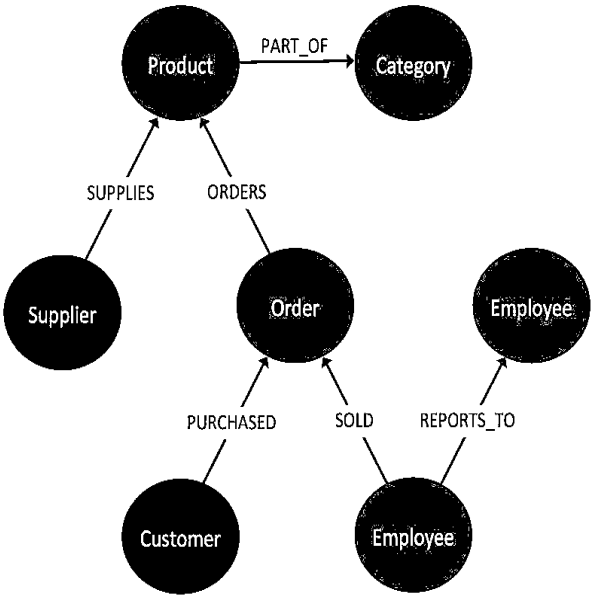


Figure 1. Example of a connected graph database model (Source: Neo4j.com)

Some of well-known graph database representatives are Neo4j, InfoGrid, Infinite Graph, Apache Graph, DEX, and OrientDB.

THE DEVELOPMENT

This section discusses in brief the development of MyRecMovie. MyRecMovie is a movie portal developed in a web environment and using graph-powered recommendation engine. Our work is inspired by Kernix Lab's work (2016) in which, graph-powered recommendation engine is proposed and developed using Neo4j by Neo4j Incorporation. Basically, the recommendation engine developed on Neo4j technologies used for storing data structured as graph and requesting it with the Cypher language.

System Overview

MyRecMovie recommendations are based on movie's genre and rating. User can click any movie, and thenthe system will display the movie's details together with a list of recommended movies based on the movie's genre that the user has clicked.

As for registered user, once the user has logged into MyRecMovie, user's related data will be retrieved from the database and displayed to the user. Registered users can rate movies and MyRecMovie will use the information to recommend movies based on the rated movies. The recommended movies will be displayed when the user views their profile or when views a movie.

Recommendation Engine Architecture

MyRecMovie is developed using Express.js framework and Node.js as a runtime environment to run the JavaScript programming language on the server side. Fig.2 shows the system architecture of MyRecMovie.

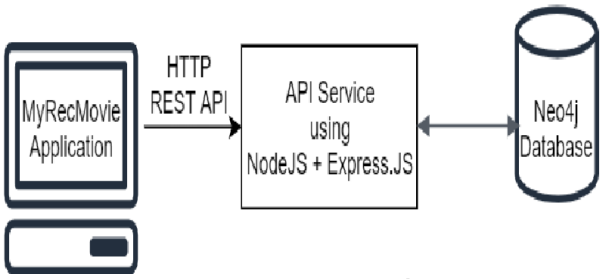


Figure 2. MyRecMovie system architecture

JavaScript programming language used in the frontend and backend of the system. As for the graph database, Neo4j is used to store all the users and movies information. Neo4j contains nodes and edges to connect and see the relationships between all the nodes. CypherQuery is used to query data in the database. The user interface was built using an open source Javascript library, React.js.

While, Gulp was used to automate the tasks developed for the frontend. In which, gulp reads files as streams and pipes the streams to different tasks. These tasks are code-based and use plugins. The tasks modify the files, building source files into production files. Gulp used to spin the web server and reload the browser.

Graph Database Model

Fig.3 shows the graph database model for MyRecMovie system. It has 5 nodes and each node can have their own attributes, such as the Genre node have the *id* and *name* attributes. Each node may have relationships with other nodes, for example, Movie node has relationship of HAS_GENRE with Genre node. When they have relationship, the recommendation engine can detect what genre that the movie have and link it with other movies that have the same genre.

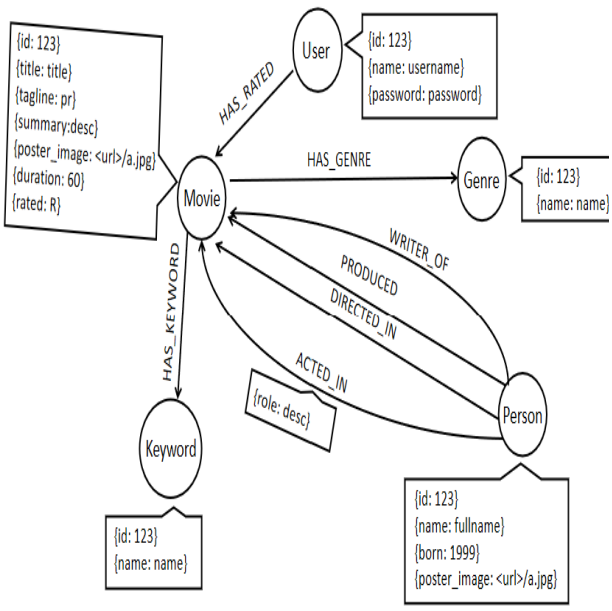
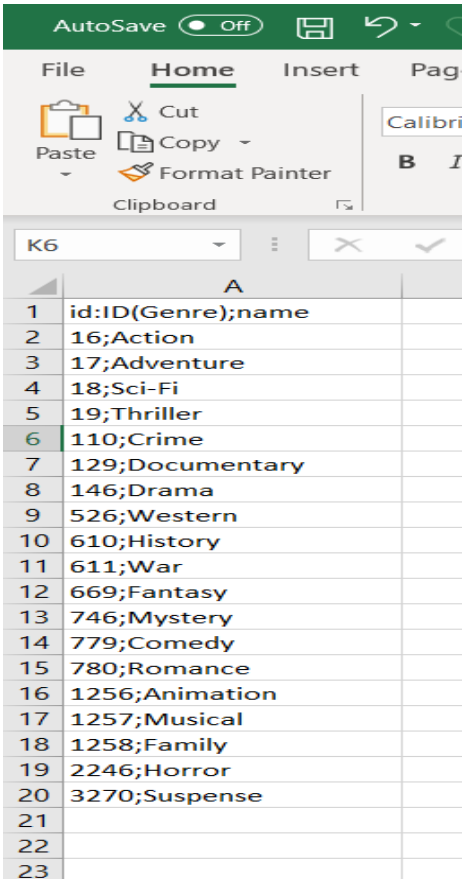


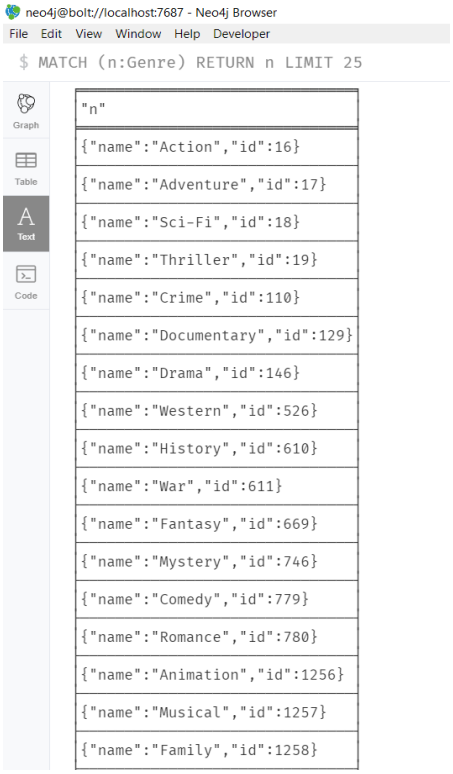
Figure 3. Graph database model for MyRecMovie system

Graph Database Implementation

The data and their relationships are created in .csv files format. Then, the files will be loaded to the Neo4j server. Fig.4 shows an example of Movie node's properties in .csv format (a) and (b) Movie node after loaded in Neo4j.



a.



b.

Figure 4.(a) .csv file view; (b) Neo4j view.

For example, Movie has Genre. Thus, this relationship is created in HAS_GENRE_RELS.csv file that contains *start_id*(Movie) and *end_id*(Genre) properties as shown in Fig. 5.

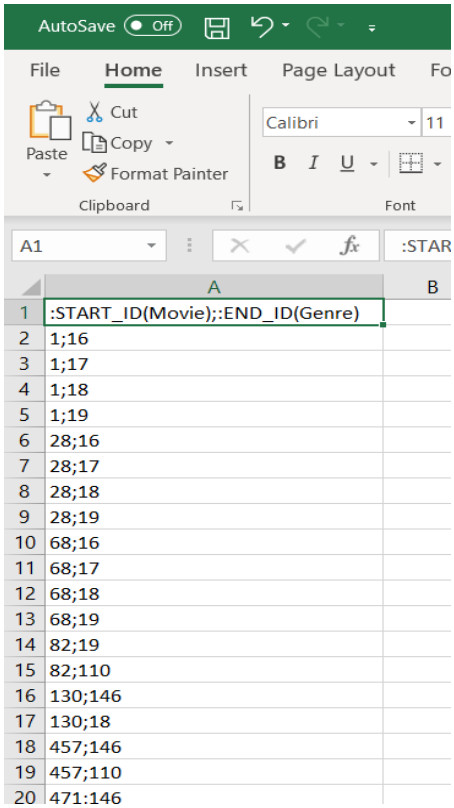


Figure 5. Relationships in .csv file view

Then the .csv file loaded into Neo4j and represented as in the screenshot in Fig. 6 that the Movie node has a relationship with Genre node and a Movie may have one or many Genre.

Figure 6. Relationships in Neo4j view

Fig. 7 is a screenshot that shows all the nodes and its relationships based on the data stored in MyRecMovie.

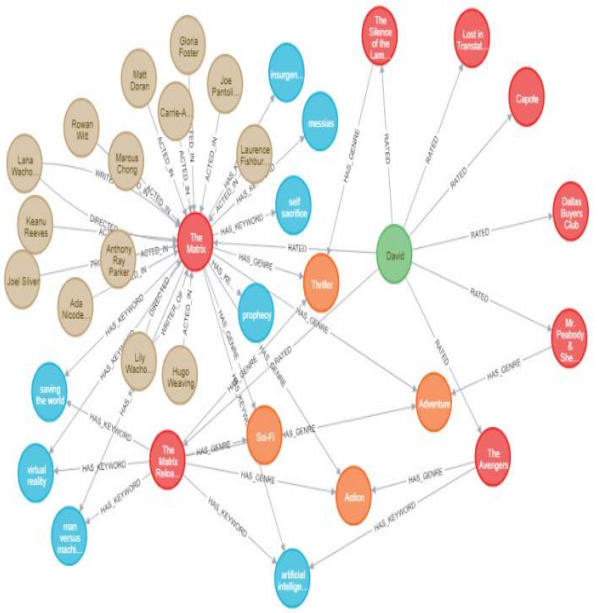


Figure 7. All nodes and its relationships in Neo4j view

RESULTS AND DISCUSSION

In summary, MyRecMovie provides recommendations based on:

- The genres of available movies by using content-based approach
- The user's previous rating by using collaborative filtering approach
- The combined ratings of all members who have similar tastes to the user by using collaborative filtering approach

Content-based Filtering

This technique analyses attributes of the movie to recommend a new recommendation to the user. This technique is applied when the user clicked on the movie, there will be a set of recommended movies that has a similar keyword to the movie that they have clicked. Fig. 8 is an example of Neo4j query.

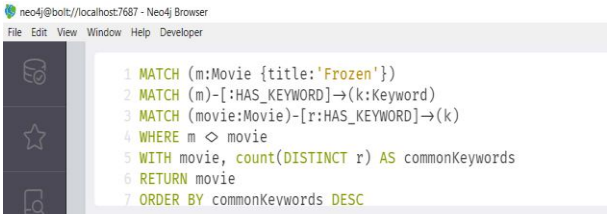
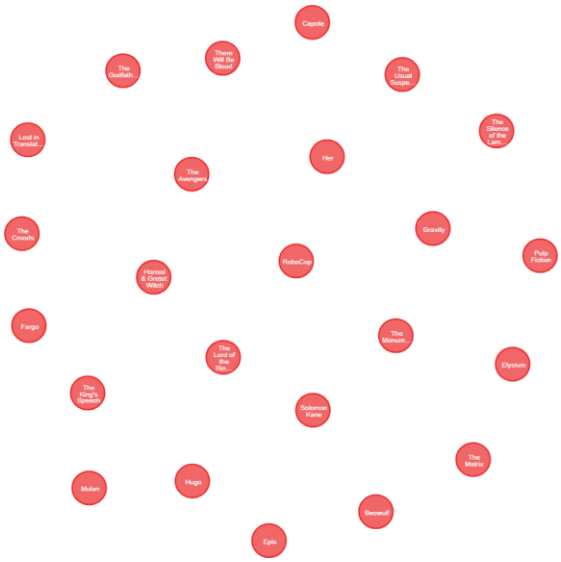


Figure 8. Neo4j query

From the query above, a set of movies will be recommended to the user as shown in the two screenshots in Fig. 9. Screenshot (a) shows the recommended movies along with its properties and screenshot (b) shows the Movie node that contains the movie name only in Neo4j view.

a.



b.

Figure 9. Recommended movies in Neo4j view

Collaborative Filtering

This technique uses a user-based collaborative method that used to find the similarity between users of the system by comparing their ratings of the same movie. An example of the Neo4j query is shown in Fig. 10.

Figure 10. Neo4j query

From the query, a set of recommended movies then generated as shown in the two screenshots in Fig. 11. The first screenshot shows the recommended movies with its properties. The second screenshot is the movie node that contains the movie's name only.

Figure 11. Recommended movies in Neo4j view

Fig. 12 shows the output displayed to the user based on data in Fig. 11.

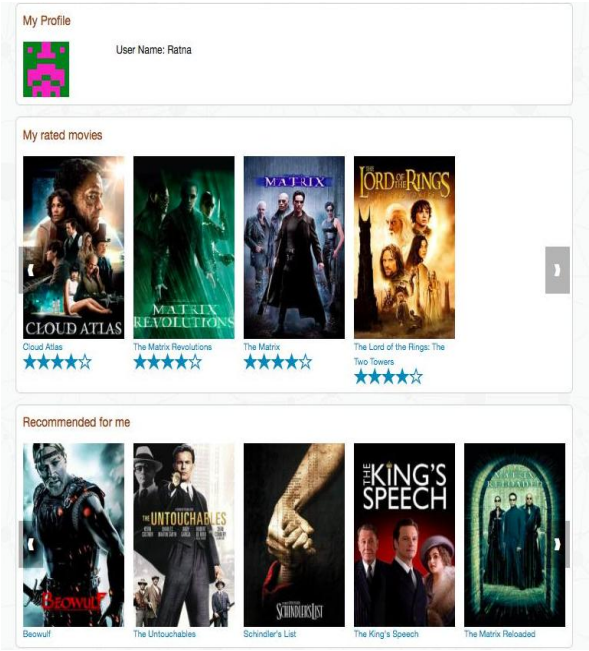


Figure 12. Recommended movies based on rating

CONCLUSION

Recommender system implemented in most e-commerce applications able to help users to lighten the problem of information overload and enables them to get recommendations of relevant products and services. This paper presents the development of graph-powered recommendation engine for a movie portal. The recommended movies are based on the movie's genre, and the user's rating of the movie and also the relationship of the movies with others rating. Graph-powered recommendation engine adoptsgraph database concept, and filtering approaches in generating the movie recommendations. Wherein, graph database has been proven that it can be used for wide and huge data in giving recommendations to the user.

However, there is room for improvement of the system presented in this paper that we will consider in our future work. Firstly, the system can be improved by providing features for the user to write reviews or commentsto the moviethat they have watched. In which, the sentiment analysis can be used to analyse the reviews in order to know what the user actually feel about the movie that they have watched. The results from the sentiment analysis can be integrated with rating data to provide better and more personalized movie recommendation to the user. By having the features, the movie portal can give more personalized recommendations to the user rather than depending on the star rating only. Secondly, the limitations of the recommendation approaches, for example, CB approach that depends on the information given by the user to predict the recommendations, can be solved by using a hybrid approach (i.e., a combination of CF approach and CB approach). Where, it predicts what the users might like separately based on the two approaches, and then combining the results to give recommendations to the user.In addition to that, knowledge-based recommender systems can be used to improve the system by considering defined preferences and match them to the corresponding items (Jan Škrášek, 2015). Conversational case-based recommenders (e.g., SHREIK(McSherry, D.*et al.* 2008)) might also useful to capture the user preferences and retrieve similar items.

ACKNOWLEDGEMENTS

We wish to acknowledge support from Universiti Putra Malaysia and industry collaborator.

REFERENCES

1. Acilar, A.M., & Arslan, A., (2009). A collaborative filtering method based on Artificial Immune Network. *ExpSystAppl* 2009, 36(4), pp.8324–32.
2. Amazon. BusinessDictionary.com. Retrieved May 08, 2019, from BusinessDictionary.comwebsite:http://www.businessdictionary.com/definition/Amazon.html
3. Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A., (2013). Recommender systems survey. *Knowledge-based systems*, 46, pp.109-132.
4. Celma, O., & Serra, X., (2008). FOAFing the Music: bridging the semantic gap in music recommendation. *Journal of Web Semantics*, 6(4), pp.250–256.
5. Chen, L.S., Hsu, F.H., Chen, M.C., &Hsu, Y.C., (2008). Developing recommender systems with the consideration of product profitability for sellers. *Int J Inform Sci* 2008, 178(4), pp.1032–48.
6. Gomez-Uribe, C. A., & Hunt, N., (2016). The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACMTransactions on Management Information Systems*, 6(4).
7. Hu, R., & Pu, P., (2009). Potential acceptance issues of personality-ASED recommender systems. In: Proceedings of ACM conference on recommender systems (RecSys'09), New York City, NY, USA;October 2009. pp. 22–5.
8. Hussain, A., Mkpojiogu, E.O.C., Kamal, F.M. (2016). Mobile video streaming applications: A systematic review of test metrics in usability evaluation. Journal of Telecommunication, Electronic and Computer Engineering, 8 (10), pp. 35-39.
9. Hussain, A., Mkpojiogu, E.O.C. (2016). Usability evaluation techniques in mobile commerce applications: A systematic review. AIP Conference Proceedings, 1761, art. no. 020049, .
10. Jalali, M., Mustapha, N., Sulaiman, M., & Mamay, A., (2010). WEBPUM: a web-based recommendation system to predict user future movement. *ExpSystApplicat* 2010, 37(9), pp.6201–12.
11. Jan Škrášek, (2015). *Social Network Recommendation using Graph Databases*, Master Thesis, Faculty of Informatics, Masaryk University.
12. Joseph Pellegrino, (2017). Flexible recommender systems based on graphs. AISR2017, May 2017, Paris, France.
13. Kernix Lab, (2016). An efficient recommender system based on graph database. https://www.kernix.com/article/an-efficient-recommender-system-based-on-graph-database/
14. Khusro S., Ali Z., & Ullah, I., (2016) Recommender Systems: Issues, Challenges, and Research Opportunities. In: Kim K., Joukov N. (eds) Information Science and Applications (ICISA) 2016. Lecture Notes in Electrical Engineering, vol 376. Springer, Singapore, pp. 1179-1189.
15. Konstan, J.A., & Riedl, J., (2012). Recommender systems: from algorithms to user experience. *User Model User-Adapt Inter* 2012:22, pp.101–123.
16. Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, pp.12-32.
17. MacKenzie, I., Meyer, C., & Noble, S. (2013). How retailers can keep up with consumers. Retrieved from McKinsey & Company: https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers
18. McSherry D., Hassan, S., &Bustard, D., (2008). Conversational Case-Based Reasoning in Self-Healing and Recovery. In: AlthoffKD., Bergmann R., Minor M., Hanft A. (eds) Advances in Case-Based Reasoning. ECCBR 2008. Lecture Notes in Computer Science, vol 5239. Springer, Berlin, Heildelberg.
19. Min, S.H., & Han, I., (2010). Detection of the customer time-variant pattern for improving recommender system. *ExpSystApplicat* 2010, 37(4), pp.2911–22.
20. Neo4j, Retrieved from https://neo4j.com/ and https://neo4j.com/business-edge/properties-the-heart-of-graph-databases/
21. Pan, C., & Li, W., (2010). Research paper recommendation with topic analysis. *2010 International Conference On Computer Design and Applications*, Qinhuangdao, 2010, pp. V4-264-V4-268, doi: 10.1109/ICCDA.2010.5541170.
22. Pathak, B., Garfinkel, R., Gopal, R., Venkatesan, R., &Yin, F., (2010). Empirical analysis of the impact of recommender systems on sales. *JManage Inform Syst* 2010, 27(2), pp.159–88.
23. Pu, P., Chen, L., & Hu, R., (2011). A user-centric evaluation framework for recommender systems. In: Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11), ACM, New York, NY, USA; 2011, pp. 57–164.
24. Resnick, P., & Varian, H. R., (1997). Recommender systems. *Communications of the ACM*, 40(3), pp.56-59.
25. Solsman, J. E., (2018, January 10). YouTube's AI is the puppet master over most of what you watch. Retrieved from CNET: https://www.cnet.com/news/youtube-ces-2018-neal-mohan/
26. Vukotic, A., Watt, N., Abedrabbo, T., Fox, D., & Partner, J., (2014). *Neo4j in action*. Manning Publications Co..