

Crowdsourcing Platform for Collecting Cognitive Feedbacks from Users: A Case Study on Movie Recommender System



Luong Vuong Nguyen and Jason J. Jung

Abstract The aim of this research is to present a crowdsourcing-based recommendation platform called *OurMovieSimilarity* (OMS), which can collect and share cognitive feedbacks from users. In particular, we focus on the user's cognition patterns on the similarity between the two movies. OMS also analyzes the collected data of the user to classify the user group and dynamic changes movie recommendations for each different user. The purpose of this is to make OMS interact intelligently and the data collected faster and more accurately. We received more than a thousand feedbacks from 50 users and did the analyzes this data to group the user. A group of the users can be dynamically changed, with respect to the selection of each user. OMS now still online and collecting data. We have been trying to enrich the cognitive feedback dataset including more than 20,000 feedbacks from 5000 users, so that the recommendation system can make more accurate analysis of user cognitive in choosing the movie similarity.

1 Introduction

The recommendation system is currently popular and applied in many industrial projects, as well as in many different domains [1]. The most common but severe feature of these recommendation systems is to understand what users are thinking and expect. As an example of a store with millions of products on display, the recommendation system relies on the explicit user behaviors (e.g., reviews and ratings) and implicit ones (e.g., clickstreams) in order to provide their customers with the most relevant product.

However, how can we assure whether the recommendations of the system yield the desired results? If the system does not understand what users think about each product, it will never be able to give them the best experience. Therefore, we consider

L. V. Nguyen · J. J. Jung (✉)

Department of Computer Engineering, Chung-Ang University, 84 Heukseok, Seoul, Korea
e-mail: j2jung@gmail.com

L. V. Nguyen

e-mail: nguyenluongvuong@gmail.com

© Springer Nature Switzerland AG 2020

H. Pham (ed.), *Reliability and Statistical Computing*, Springer Series
in Reliability Engineering, https://doi.org/10.1007/978-3-030-43412-0_9

that understanding the opinions and feedbacks of the users is important and essential so that the system can understand and make suggestions that are accurate and suitable for their needs.

In particular, we focus on the movie domain, and we realized that there are many ways to get feedback from users who have a lot of experience in watching movies. Most of movie recommendation systems (e.g., IMDB, MovieLens, Netflix, and so on) collect information by asking users to rate movies and to write an opinion about personal feelings. However, many users do not want to write their conclusion carefully because it takes their time. Some of the users do not know how to write their opinion clearly and thoroughly.

We want to have a new approach to collect cognitive feedbacks from users by understanding their cognition in selecting similar movies. Then, we can recognize and predict the similar movie selection patterns of the users. Once we have such a large amount of feedback, we can use the statistical analysis methods to understand why users think these two movies are similar? To answer that question, we aim to build a crowd sourcing platform [2] to collect a large number of datasets on user feedback, as mentioned above. We propose a crowd sourcing-based recommendation platform (called OurMovieSimilarity¹) and the movie data to create sample information for user select have collected automatically by using IMDB open database, or users add themselves to the OMS through personally suggest function.

The problem is that the system has to interact with users efficiently and to be easy to use. We decided that the OMS's layout and the process must be exciting and straightforward [3]. The data collection process described as follows:

- Firstly, the system's new users will choose a movie they have watched and favorited.
- Second, OMS will suggest a list of 5 similar movies (calculated by the system and accompanied by an α parameter that is the user's selection trend, this parameter has initially defaulted 1).
- In the final step, users will select a movie in 5 movies appearance that they think it is similar to the one they chose at first step.

The movie selection will be a loop with variable ω parameters dynamic closer to the user's selection trend. The calculation to suggest a movie and the addition of ω parameters helps users improve the time to choose the movie similar, and they do not have to do it multiple times to find a similar of a movie m_i and a movie m_j . The data collected will be applied in movie similarity recommendation, as we described in Sect. 4.2.

The outline of this chapter includes five sections. Section 1 is for the introduction, and Sect. 2 discusses the related work on crowd sourcing platforms. In Sect. 3, we address a cognitive approach in the recommendation system. We describe experiences on OurMovieSimilarity platform include the overview, functions of OMS, and the data analysis in Sect. 4. Finally, in Sect. 5, we conclude and make some remarks.

¹<http://recsys.cau.ac.kr:8084/ourmoviesimilarity>

2 Related Work on Crowdsourcing Platforms

In 2006, in the article “The Rise of Crowdsourcing” which Jeff Howe published in Wired [4], the term *crowdsourcing* appearance the first time, which is a contraction of *crowd* and *outsourcing*. This quote from the article describes simply what crowdsourcing is and how it’s made possible by technological advances. The meaning of the word crowdsourcing itself is a portmanteau of crowd and outsourcing [5, 6]. From an online application perspective, online crowdsourcing platforms are increasingly being used to capture ideas from the crowd. Global companies are adopting crowdsourcing ideas to connect with and get feedback from the users. The success of a crowdsourcing platform largely depends on members and their motivation to participate. Motivation determines the quality and quantity of contributions [7].

Crowdsourcing has become a useful tool for understanding audience preferences and anticipating needs [8], which are very important for businesses that depend on innovating or enhancing their products such as fashion brands, food manufacturers, or restaurants [2]. For example, Starbucks provides a simple fill form where people can suggest new ideas, enhance existing services, or request product deliveries. More complex, Amazon MTurk and CrowdFlower [9], which are crowdsourcing platforms, have several jobs such as data categorization, metadata tagging, character recognition, voice-to-text transformation, data entry, email harvesting, sentiment analysis, ad placement on videos, or surveys.

ResearchGate [10] was founded in 2008 by Ijad Madisch, who aims to transform the way researchers are doing their research [11]. For the first time, ResearchGate’s headquarters started in Boston and is currently based in Berlin, Germany, and invested by several US venture capital companies. Researchgate now has more than 7 million members., with an average of seven researchers signing up per minute. This system is a crowdsourcing platform for scientists and researchers to ask, answer the question, finding collaborators, and share papers. Although reading an article does not require registration, those who want to become a website member need to have an email address at an accredited or manually verified organization that is a published researcher to register. Each member of the site has a user profile and can upload research results, including papers, data, chapters, adverse effects, patents, research proposals, methods, presentations, and software source code. Besides, users can also follow the activities of other users and participate in discussions with them. Users can also block interactions with other users.

ResearchGate’s success has allowed researchers to disseminate their ideas and share their publications for free so they can create collaborations between researchers from around the world. Through this crowdsourcing platform, members can maintain their publications, ask and answer research-related questions, and follow other researchers to receive their publication updates. This system is capable of tracking the activities of researchers identified as related to each other (such as co-author of the article, project, chapter, and so on.) to send automatic notification emails about updates that associated members. And in fact, this system also has ways to summarize the joint activities of a research group to assign a unique email to limit the number of duplicate emails sent to each researcher.

3 A Cognitive Approach to Recommendation

The first question we asked was, “*why users think two movies are similar?*”. Recent advances in psychology and cognitive sciences support the notion that people use a dual-process model, whereby a sense of similarity built on a combination of feature-based taxonomic and relationship-based thematic relations [12]. Taxonomic or hierarchical relationships based on internal characteristics, such as the features of the items themselves, while a thematic relationship is external, which is a separate event or scene that connects the two items. For example, pen and pencil are taxonomically similar since they share many features: both using for writing and under the category of “stationery”. Pen and papers are thematically similar since they often used during the same event.

Nowadays, people have witnessed a massive increase in electronic service, e-commerce, and e-business applications operating on the Internet [13]. Recommended systems are increasingly being used by application vendors to make recommendations to their customers [14]. However, most traditional recommendation systems primarily focus on extracting and recommending general priorities based on historical user data [15]. Although users, in general, shared interests can be considered relevant, an individual user also has his personal preferences. He/she can also answer domain expert knowledge to some extent to make a decision. Moreover, very often, while using traditional recommendation systems, users do not readily distinguish whether the items on the page are actual recommendations or merely the content of the page is displayed indiscriminately for all users. Therefore, traditional recommendation systems do not give customers the feeling of being treated individually. Jeff Bezos, CEO of Amazon.com, concluded that if I had 3 million customers on the Web, I should have 3 million stores on the web [15]. Personalized proposal agencies are emerging to overcome the personal nature of integrated recommendations by using technology to assist customers in making decisions about how to treat each customer individually [16].

4 Experiences on OurMovieSimilarity Platform

4.1 Overview

We create OurMovieSimilarity (OMS) platform, which is the crowdsourcing system to collect similar movies from a large amount of users’ cognitive data in the selection of similar movies. OMS collects data of movies from the data warehouse provided by IMDB and allows users to have the experience of choosing similar movies from the movies they have seen and also their favorite ones. Figure 1 shows the main features of the OMS system: data collection from the internet and processing of data obtained from users’ cognitive of similar movies.

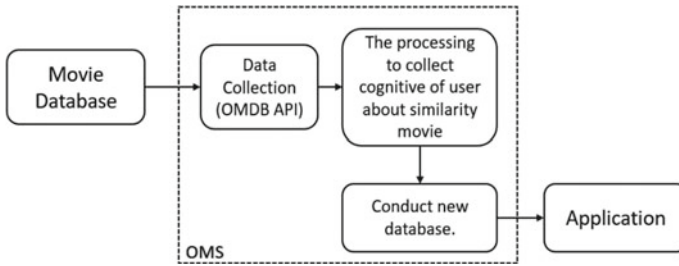


Fig. 1 The overview of the OMS crowdsourcing platform

OMS system is built based on Java programming language, combined with the MySQL database. OMS contains two services, which are web service and background service. The system designed by the Model-View-Controller (MVC) model to ensure security and many flows to handle multiple access and tasks with extensive data. We implement the web service side with Apache Tomcat Servlet Container and Java Server Pages technology. For connecting two services, we choose the Apache Thrift framework. The main features of the OMS system: data collection from the internet and processing of data obtained from users' cognitive of similar movies. Specifically, we have two methods of collecting data about movies, first is to collect automatically from the IMDB database, users add second. Then, OMS collects data from users' selection about the similarity of the movie. Finally, we have a new data set of user perceptions after performing the analysis of the collected data.

The primary purpose of OMS is focussing on simple and efficient. However, OMS is a web-based crowdsourcing platform, and therefore, the problem of loading data from the system to display to users must ensure the lowest latency. Besides, a big problem for a web-based application is making sure that the user has enough instructions to operate the entire system. To solve the issues which we mentioned above, we apply the concept of progressive disclosure to OMS: "show users what they need when they need, and where they want" is the main idea of it. During this period, we focused on increasing the amount of feedback from users most efficiently and accurately. To keep the system running smoothly, and improve the user experience, we take an approach for users to interact simplest and receive the fastest response from the system.

Notably, we apply one template for all pages to maintain the consistency of the user interface. Hence, similar signs will be easily recognized by users (e.g., buttons, functions) while they operate on OMS. We are also applicable this thought to all actions occurring on the system. That is, with all the different steps, we all guarantee the same interface behavior. All of the features we mentioned above based on three gold rules of user interface design [17].

4.2 How Can OMS Interact with Users Intelligently

After users select movies which they have seen and favorite, OMS will calculate and suggest the list of 5 movies similar to their selection. Given m_1 is a movie which users have seen and m_2 is one of all movie remain in our database, we aim to calculate score $Sim(m_1, m_2)$ by making a score after comparing each feature of two movies. The formula for scoring $Sim(m_1, m_2)$ described as follows:

$$Sim(m_1, m_2) \equiv \langle T, G, D, A, P \rangle \quad (1)$$

where T, G, D, A , represents the feature of comparing between title; genre; director; actors; and plot of a movie m_1 and a movie m_2 . We repeat that calculation for all the remaining movies in the OMS and obtained a set $\{Sim(m_1, m_2); Sim(m_1, m_3); \dots; Sim(m_1, m_n)\}$. Then, we add an element that is the trend of users while they select similar movies. Hence, the formula has rewritten as follow:

$$Sim(m_1, m_2) = \frac{\sum_{i=1}^n \omega_i \times Sim_i(m_1, m_2)}{\sum_{i=1}^n \omega_i} \quad (2)$$

where ω_i are user's trend of selecting similar movies based on i which a number of features: title, genre, director, actors, and plot, and Sim_i is a score between movie m_1 and movie m_2 based on each feature. To perform the making score of Sim_i after comparing the similarity of each feature, we use a string comparison algorithm base on the Jaccard similarity coefficient (originally coined coefficient by Paul Jaccard) [18]. We do the calculation by identifying the union (characters in at least one of the two sets) of the two sets and intersection (characters which are present in set one which is present in set two). To generalize the calculation formula, we define the following:

$$Sim_X(m_1, m_2) = \frac{|X_{m_1} \cap X_{m_2}|}{|X_{m_1} \cup X_{m_2}|} \quad (3)$$

where X is a feature of each movie m_1 and movie m_2 . In particular, the score of titles features and described as follows.

$$Sim_T(m_1, m_2) = \frac{|T_{m_1} \cap T_{m_2}|}{|T_{m_1} \cup T_{m_2}|} \quad (4)$$

where T_{m_1} represents the title of the movie m_1 , and T_{m_2} represent the title of the movie m_2 . We perform this calculation for the features: title, genre, director, actor. In case of making a score by comparing the plot of the movie m_1 and movie m_2 . Because, it is not a regular string to compare, we pre-processing first. So, we present Term Frequency–Inverse Document Frequency (TF-IDF) to select high weight terms from the plot of

the movie. First, we count the number of times each word occurs in the plot of the movie following the definition of Hans Peter Luhn (1957): “The weight of a term that occurs in a document is simply proportional to the term frequency” [19]. Number of times that term occurs in the plot of movie denoted by raw count. Second, because some term *the*; *a*; *an*, and so on are common and tend to incorrectly emphasize scenario which happens to use this term more frequently without giving enough weight to the more exact another time. Hence, following the definition of Karen Sparck Jones: “The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs” [20] we use inverse document frequency (IDF) measure how much information the term provides.

Finally, when we use TF-IDF combining with the defined threshold for filtering the useless term ∂ , we can find the high weight term frequency in the plot of the movie. So, the formula is shown as follows:

$$P_{m_i} = \left\{ t \mid f(t, p_{m_i}) \times \log \frac{N}{|\{p_{m_i} \in M : t \in p_{m_i}\}|} \geq \partial \right\} \quad (5)$$

where P_{m_i} represent all high-frequency term t in a plot of the movie m_i ; t is a term that occurs in the plot of a movie m_i ; p_{m_i} is all term in the plot of a movie m_i ; N is the number of total plots; M is the whole term in all plots remaining; ∂ is the defined threshold for filtering useless term.

By applying TF-IDF, we have obtained for each film a plot of high-meaning terms. So that the comparison between two similar films according to plot feature has more accuracy. Of course, this function would be done entirely automatically, and it means when we have a new movie added to the OMS system, all data, when retrieved, are recalculated to meet the highest accuracy. Beside, OMS will auto dynamic show the suggest movies to users for selection based on the trend of their choice. For example, when user select movie m_1 is “have seen”. In the first time, we use our calculating to show movies for users’ selection, and the OMS will show a set $\{m_i \mid i \in [2, \dots, 6]\}$, and users perhaps have some refresh to select a movie similar. However, after some round limited, we can predict the trend of users’ selection described in Fig. 2; and when a user select movie m'_1 is “have seen” we will show $\{m'_i \mid i \in [2, \dots, 6]\}$ for a user to select a movie that is more similar than m'_1 . At this time, showing movies will be dynamically calculated and displayed depending on the users’ selection trend. Hence, a user quickly to select similar movie after changing suggested movies one or two times.

Considering the aspects of the relevance of the users in our system, we group related users through the analysis of each user’s data. Suppose, users U_i and U_j focus on the priority features to choose the movie similarity in the order $\langle Title, Genre, Actor, Director, Plot \rangle$; the system identifies these two users related. Otherwise, the user U_i focuses on $\langle Title, Genre, Director, Actor, Plot \rangle$ and U_j focuses on $\langle Title, Genre, Plot, Director, Actor \rangle$, the system thinks that these two users are likely to be related together. In this case, the system will calculate and assesses the relevance of these two users for grouping. Therefore, we use soft cosine to measure the similarity between

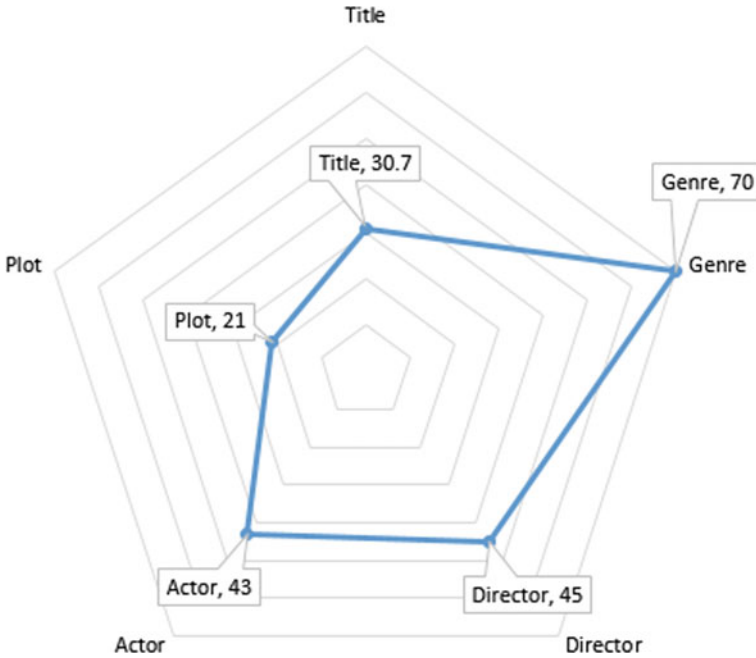


Fig. 2 The trend of the user’s selection

two users. The formula to calculate the related between two users as following:

$$Sim(U_1, U_2) = \frac{\sum_{i,j}^F s_{ij} U_{1i} U_{2j}}{\sqrt{\sum_{i,j}^F s_{ij} U_{1i} U_{1j}} \sqrt{\sum_{i,j}^F s_{ij} U_{2i} U_{2j}}} \tag{6}$$

where s_{ij} is the similarity of each feature between two users; F is the number of user’s features. From that assumption, we do user grouping and use the data of users in the same group to recommend their chosen movie pairs. For example, user U_i selects the movie (m_i, m_j) the same and user U_i is related to user U_j , the system will suggest that user U_j respond to pair of movie (m_i, m_j) whether or not? The user grouping is shown in Fig. 3.

4.3 Statistical Analysis of the Data (Cognitive Feedbacks)

There are many sources to gather information about movie data provided online. However, we identify an IMDB is an extensive and highly scalable movie database. We collected over 20,000 popular movies from 1990 to 2019 with nine genres and continue importing from open movie database IMDB, 3439 directors, and 8057 actors. The statistics of movie data in OMS is described in Table 1.

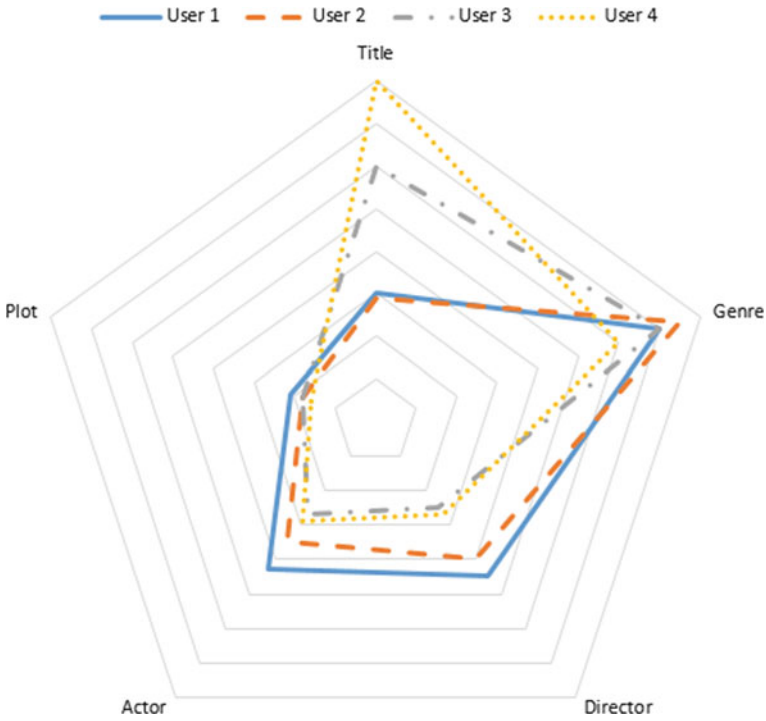


Fig. 3 The related of users based on their trend in selecting movie similarity

Table 1 Statistics of movies data in OMS

#	Title	Amount
1	Number of movies	20,000
2	Range of releases	1990–2019
3	Number of genres	9
4	Range of directors	3439
5	Range of actors	8057

Our system now online and continue collecting data, at this time, we have over 50 users and already raised more than 1000 pairs of movie similarity from users’ activities.

The number of data collected from users is shown in Fig. 4, and has the format: $(U_i, m_j, m_k, Sim(m_j, m_k), \vartheta_i)$ inside U_i is the id of the user; m_j and m_k are a pair of the movie similar; $Sim(m_j, m_k)$ is a similarity score we calculated; and ϑ_i is the number of times user change suggested movies in select a pair of movies similar.

We have also done user grouping, but with the number of users in this experiment, the user grouping is too fragmented. We aim to reach a large number of users so that clustering will enhance more efficiency. Figure 5 shows a graph representing several typical user groups in our system.

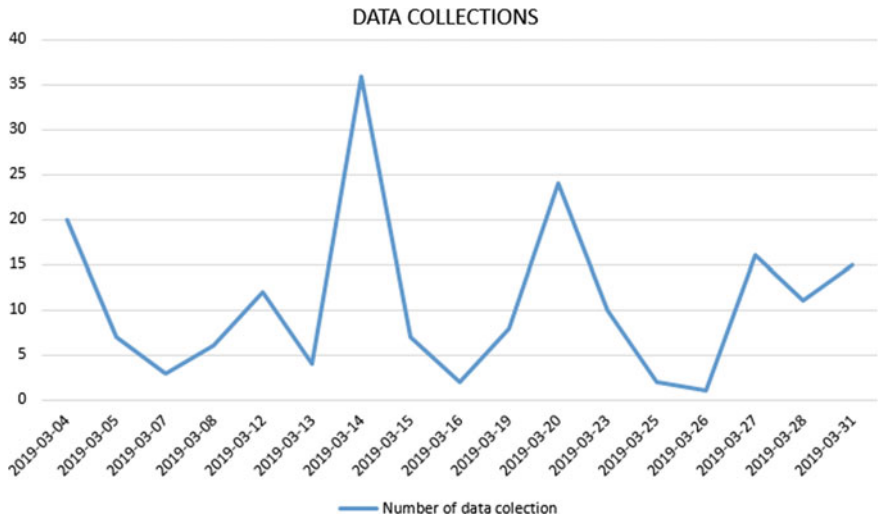


Fig. 4 Statistics amount of data feedback from users by day

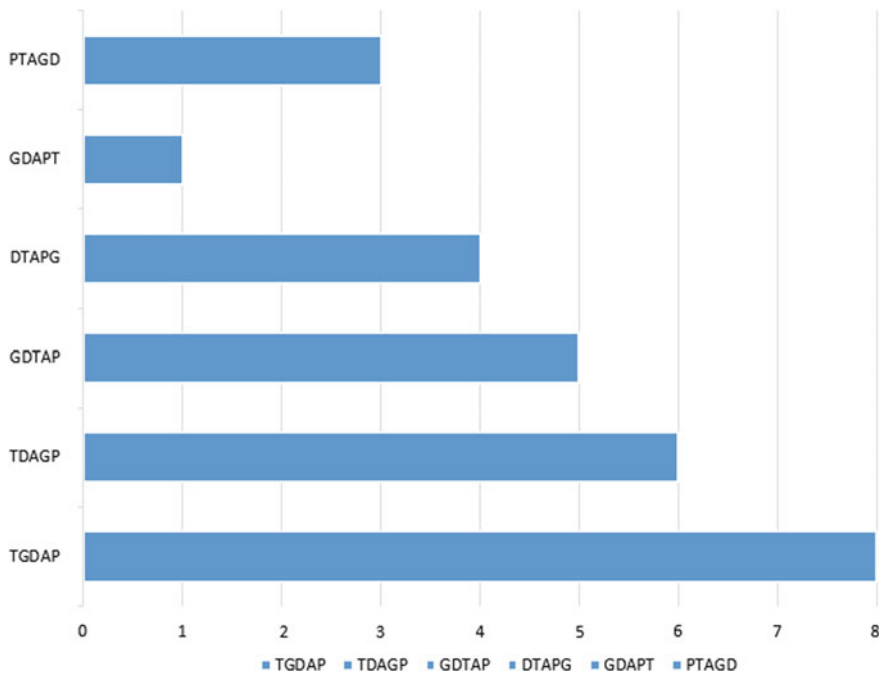


Fig. 5 Clustering of users with T, G, D, A, and P represented sequence title; genre, director; actor; and plot

5 Concluding Remarks

Our study focuses on understanding the cognitive of users and answers the question, “what makes users think two movies are similar?”. To do that, in the first stage, we deploy OurMovieSimilarity as a crowdsourcing platform to collect cognitive data from a crowd of users.

Besides, OMS has performed the calculation of data collected by each user to analyze and predict the trend of the user’s selection. The data collected shows that the number of page refresh in choosing a pair of movies similarity reduced for each user. This helps to minimize the time, boring, and increase the ability to collect user feedback. From the approach to interacting with such a user, we deduce that people will be motivated to contribute feedback to help others, as well as discover for themselves.

OurMovieSimilarity is still deploying online and continuing to collect data. We plan to have at least 5,000 users and 20,000 responses to make more accurate analyzes of user cognitive in choosing the movie similarity. We release dataset representing user perception of similarity between two movies at the OMS website.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2017R1A2B4010774).

References

1. Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook. In: Recommender systems handbook. Springer, pp 1–35
2. Zheng H, Li D, Hou W (2011) Task design, motivation, and participation in crowdsourcing contests. *Int J Electron Commer* 15(4):57–88
3. Baturay MH, Birtane M (2013) Responsive web design: a new type of design for web-based instructional content. *Procedia-Soc Behav Sci* 106:2275–2279
4. Howe J (2006) The Rise of Crowdsourcing. *Wired Mag* 14(06):1–5
5. Schenk E, Guittard C, Others (2009) Crowdsourcing: what can be outsourced to the crowd, and why. In: Workshop on open source innovation, Strasbourg, France, vol 72, p 3
6. Estellés-Arolas E, González-Ladrón-de-Guevara F (2012) Towards an integrated crowdsourcing definition. *J Inf Sci* 14
7. Janzik L (2010) Contribution and participation in innovation communities: a classification of incentives and motives. *Int J Innov Technol Manag* 7(03):247–262
8. Gadiraju U, Kawase R, Dietze S, Demartini G (2015) Understanding malicious behavior in crowdsourcing platforms: the case of online surveys. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems, pp 1631–1640
9. Difallah DE, Catasta M, Demartini G, Ipeirotis PG, Cudré-Mauroux P (2015) The dynamics of micro-task crowdsourcing: The case of amazon mturk. In: Proceedings of the 24th international conference on world wide web, pp 238–247
10. Yu M-C, Wu Y-CJ, Alhalabi W, Kao H-Y, Wu W-H (2016) ResearchGate: An effective altmetric indicator for active researchers? *Comput Human Behav* 55:1001–1006
11. Dolan KA (2012) How Ijad Madisch aims to disrupt science research with a social network. *List Forbes* (2012)

12. Wisniewski EJ, Bassok M (1999) What makes a man similar to a tie? stimulus compatibility with comparison and integration. *Cogn Psychol* 39(3–4):208–238
13. Kalakota R, Robinson M (2001) *e-Business 2.0: Roadmap for success*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©2001
14. Ben Schafer JRJ, Konstan J (1999) Recommender systems in E-commerce. In: *Proceeding of ACM conference on electronic commerce (EC-99)*, pp 158–166
15. Ben Schafer J, Konstan JA, Riedl J (2001) E-commerce recommendation applications. *Data Min Knowl Discov* 5(1–2):115–153
16. Häubl G (2001) Recommending or persuading? the impact of a shopping agent’s algorithm on user behavior. In: *Proceedings of the 3rd ACM conference on electronic commerce*, pp 163–170
17. Mandel T (1997) *The elements of user interface design*, vol 20. Wiley, New York
18. Bouchard M, Joussemle A-L, Doré P-E (2013) A proof for the positive definiteness of the Jaccard index matrix. *Int J Approx Reason* 54(5):615–626
19. Luhn HP (1957) A statistical approach to mechanized encoding and searching of literary information. *IBM J Res Dev* 1(4):309–317
20. Jones KS (2004) A statistical interpretation of term specificity and its application in retrieval. *J Doc*

Luong Vuong Nguyen is a Ph.D. Student in Chung-Ang University, Korea since August 2018. He received the B.S. in Department of Mathematics from Da Nang University of Education and Science, Vietnam in July 2009. And he received M.S. degrees in Department of Computer Engineering from The Da Nang University of Technology in Vietnam in April 2013. His research topics include knowledge engineering on data science by using data mining, machine learning, ambient intelligence, and logical reasoning.

Jason J. Jung is a Full Professor in Chung-Ang University, Korea, since September 2014. Before joining CAU, he was an Associate Professor in Yeungnam University, Korea since 2007. Also, He was a postdoctoral researcher in INRIA Rhone-Alpes, France in 2006, and a visiting scientist in Fraunhofer Institute (FIRST) in Berlin, Germany in 2004. His research topics are knowledge engineering on social networks by using many types of AI methodologies, e.g., data mining, machine learning, and logical reasoning. Recently, he have been working on intelligent schemes to understand various social dynamics in large scale social media.