An Implementation of Movie Recommender System Using Content-Based Filtering

Cheng-Yuan Ho[1], Edo Novanto[2]
[1]Dept. of Computer Science and Information Engineering, Asia University, Taichung City, Taiwan
[2]Dept. of Electronic and Informatic Engineering, Gadjah Mada University, Yogyakarta, Indonesia
tommyho@asia.edu.tw, edonovanto@mail.ugm.ac.id

Abstract— "Data is new oil" is the term used to express the importance of data usage. There are various kinds of data usage, and one of them is to make a recommendation system. The recommendation system, such as e-commerce, music, books, places, and movie, becomes more and more important in the industrial world because the recommendation system helps users to find things that might interest them. This paper aims to provide an explanation of how a movie recommender system works with content-based filtering methods and describe the implementation of the movie recommender system.

***Index Terms*—data usage, movie, film attribute, recommender, recommendation system, content-based filtering.**

I. INTRODUCTION

Recommendation system is a calculation that is made to be able to predict information that suits the needs and interests that will be offered to users. This system collects and calculates user data and user behaviors to improve accuracy. The applications of recommendation systems are widely used today, such as books, films, music, video, movies, and research works.

The use of system recommendations helps companies to increase user satisfactions for the products they offered. Products that are not in accordance with the wishes of the user have a major impact on the use of these products in the long run. Often, if the product offered is in accordance with the wishes of the user, then on another occasion; this user may do a very similar pattern to get satisfaction. Therefore, a good recommendation system will greatly affect the level of user satisfactions in the long period.

Making a recommendation system has to look at the background of the product users and its relationship to the way that product is used previously. For example, in the film recommendation system, people's tastes can change at any time so that the previously formed model is no longer relevant to these users. This is what underlies why it is necessary to improve the recommendation system that is able to overcome the possibility of problems that happened in the future.

The work is supported by the Ministry of Education of Taiwan, the Artificial Intelligence Talent Cultivation Project, 2018 and partially supported by the project of Asia University, called "Research on Applying the Big Data Analytics of Electronic Ticket to Construct Models of Public Transportation System Services Integration Based on the Trip Information," under Grant no. 107-asia-11.

In this article, the authors aimed to develop and implement a recommendation system that is capable of predicting and recommending films that users may like. A content-based filtering (CB) method with attention to the film attributes (director, year, actors, genre, etc.) is used in the recommender system in this article because CB helps in calculating the presentation of user interests in the film.

II. OVERVIEW

A. Recommendation Systems

In [1], there are lots of explanations for recommendation systems. Some examples of [1] are described as follows.

- A recommendation system is a computer program that recommends some sort of resource based on algorithms that rely on some sort of user model, some sort of content model, and some means of matching the two.
- A recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" that a user would give to an item.
- Recommender systems in e-commerce identify a similarity in the preferences or tastes of one consumer and others (e.g., goods purchased, products viewed); and make recommendations for new purchases drawn from the set of other goods bought or viewed by each of the like-minded consumers.
- An AI system that recommends various products, services, articles, or social connections to a user based the user's profile.

We can found that no matter which kind of description of a recommendation system, the most important goal of a recommendation system is to predict or/and recommend relevant items to a user based on his/her historical data (e.g., his/her past behavior), similar decisions made by other users, or an item with similar properties of the item he/she chose before.

B. Types of Recommendation Systems

A lot of recommendation systems are used in the real world, and some are very popular, such as Netflix, Youtube, Spotify, Facebook, Twitter, and Amazon. Furthermore, the methods of these recommendation systems from subclass to core technologies are respectively clustering, unsupervised learning, and machine learning as shown in

Figure 1 [2]. The approaches of recommendation systems can be divided into six classes: Collaborative Filtering Systems (CF), Content-Based Filtering Systems (CB), Multi-Criteria Systems (MC), Risk-Aware Systems (RA), Mobile Systems (MS), and Hybrid Systems (HS). Each approach has its own advantages, strengths, and weaknesses that is described as following paragraphs.

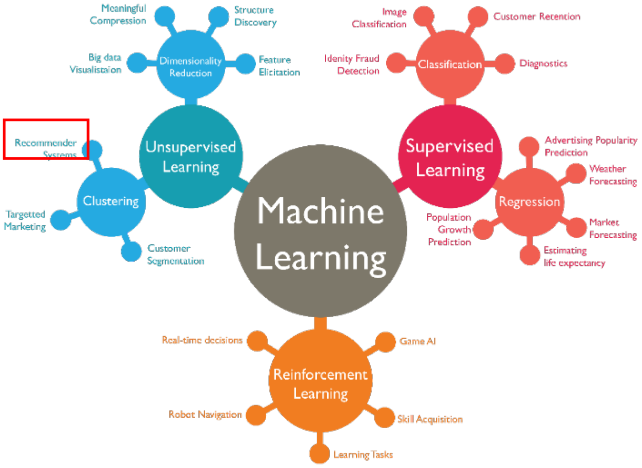

Figure 1. The core technologies of a recommender system, from subclass to core: clustering, unsupervised learning, and machine learning

1) *Collaborative Filtering Systems:*
These systems conduct an analysis of other users who have similarity track record that is likely to behave the same in the future. For example, two users have similar interests. When one gives a good rating on film A, it is likely that other user will give a good rating on film A as well. In addition, there are two types of collaborative filtering, namely user-based filtering and item-based filtering.

2) *Content-Based Filtering Systems:*
These systems provide recommendations that are similar to movies that the user has liked before. Content-based filtering is not only based on rating values, but also considers more on the descriptions and features of an item, e.g., the film's information such as title, year, actor, genre, and director.

3) *Multi-Criteria Systems:*
As the name implies, instead of using a single criterion value for the base reference, these systems incorporate preference information upon multiple criteria. In practice, they will first exploit preference information on multiple criteria affecting the overall preference value, and then predict a rating for the unexplored item.

4) *Risk-Aware Systems:*
Some items are with high risk, so it is important for a recommendation system to consider the risk level as the criterion value. In addition, some risk-aware systems incorporate risk levels into their processes.

5) *Mobile Systems:*
Due to the mobile devices, some recommendation systems deal with mobile data, such as GPS and IP address, as the reference criteria to offer context-sensitive, personalized recommendations. This kind of systems is called the mobile system.

6) *Hybrid Systems:*
A hybrid system can be thought as a combination of at least two above approaches. All approaches are calculated separately, and then, the results of these approaches are put together to get better accuracy.

We can easily find the differences between CF and CB by visualizing the approaches of CF and CB in Figure 2 [3], respectively.

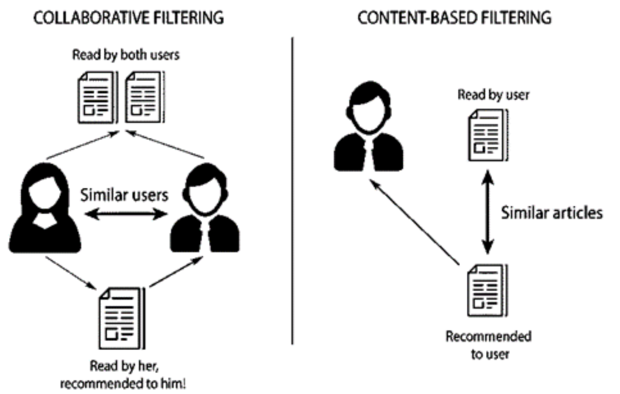

Figure 2. Visualizations of CF and CB approaches

III. RELATED WORK

Lots of recommendation systems have been developed over the years. They used one kind of approaches described in the above section, CF, CB, MC, RA, MS, and HS. Also, these systems have been implemented by various big data and machine learning algorithms.

In [3], the authors described the implementation of their recommendation system via two collaborative filtering methods by Apache Mahout and also focused on analyzing data to gain insights into the movie dataset with Matplotlib libraries in Python.

A content-based movie recommender system in [4] is proposed to capture the temporal user preferences in user modeling and predict the preferred movies. It provided a user-centered framework incorporating content attributes of rated movies for each user into a Dirichlet Process Mixture Model, and then inferred user preferences and provided a proper list. In practice, the authors implemented the system with the MovieLens dataset.

The authors used the user project scoring matrix which was pre filled and the traditional user-based collaborative filtering algorithm to implement their recommender system [5]. At the same time, the database technology is used graph database in order to deal with complex relations.

In [6], the authors combined collaborative and content-based approaches in a hierarchical manner to implement a personalized meta-level hybrid recommendation system to suggest movies to users.

A fully content-based movie recommendation system is proposed to recommend movies in [7] by making the use of a neural network with the content information of the movies to obtain features and learn the similarities between movies. Accordingly, the movies are recommended based on the similarity between them.

A recommendation system in [8] is implemented by combining user based and item-based collaborative filtering approaches. In practice, this system used nearest neighbors machine learning and unified user based and item-based recommendations.

IV. IMPLEMENTATION

This section will describe how a content-based filtering approach is implemented into a movie recommender system. There are three processes of this system: "select attributes," "establish/update user profile," and "score films' values."

1) *Select attributes:*
   Determine which attributes that will be suitable for being used in the calculation and system. Besides, in order to increase the accuracy, the film attributes are also selected for the input of the next process.
2) *Establish/update user profile:*
   Based on a user's information such as his/her rating and favorite history, and selected film attributes, the system establish/update this user profile.
3) *Score films' values:*
   For every user, the system scores all films in its dataset based on the user profile, and then the system will generate a recommended movie list to this user.

Other important elements and components are described in the following paragraphs.

- Dataset
  The dataset of the proposed system was taken from MovieLens [9]. The MovieLens is free and for non-commercial use to everybody. Also, it's a web-based recommender system and virtual community that recommends movies for its users. The dataset of the proposed system collected all data and information of dataset in MovieLens. In other words, it included 100,836 number of ratings given by 610 users on different films and a collection of these users' information.
- Data Cleaning
  The proposed system looked for users who give the most ratings since, with this action, it would get a better accuracy and avoid User Cold Start. Data from the user will be divided into training data and testing data. Furthermore, the proposed system would add additional attributes such as director, actor, and IMDb rating [10] by OMDb API [11] for every film as shown in Fig. 3.
- Data Calculation
  The proposed system calculated each attribute to get the user value by counting the number of each attribute seen by the user having the interest in that attribute.

UserId	Rating	ImdbID	Title	Genre	Director	Actor	imdbRating

Figure 3. Attributes for traing data and testing data

An example is described in the following paragraphs and shown in figures 4~6. First, from OMDb API, the system can get all films' attributes as Fig. 4. Next, as shown in figure 5, when a user watches a film, the system will add this film's attributes into this user's film list, and then according to the rating given by the user, it will calculate the corresponding scores for the user's interest/suka, and save the scores to the user's user profile. Finally, based on the user profile and films' attributes, the system can calculate and recommend the film with the highest score to the user's suka as Fig. 6.

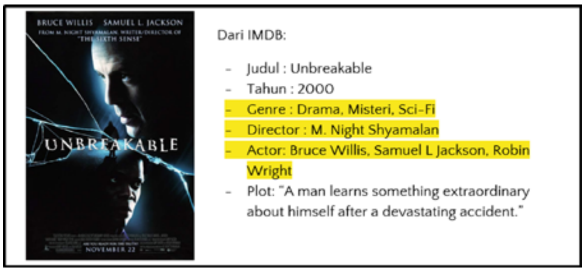

Figure 4. A film's attributes, where the words in yellow are important

	Mystery	Sci-Fi	Drama	Thriller	Horror
Unbreakable	1	1	1		
The Sixth Sense	1		1		1
Gone Girl	1		1	1	

User A	3	1	3	1	1

(a) The calculation example with default value (no rating scores)

	Mystery	Sci-Fi	Drama	Thriller	Horror	Rating
Unbreakable	1	1	1			9
The Sixth Sense	1		1		1	6
Gone Girl	1		1	1		7.5

User A	22.5	9	22.5	7.5	6

(b) The calculation example when a user gives rating scores

Figure 5. The proposed system calculates the corresponding scores for a user's interest and saves the scores in user profile

	Mystery	Sci-Fi	Drama	Thriller	Horror	Suka
Angel & Demons	1			1		???
The Ring	1				1	???
Arrival	1	1	1			???

User A	22.5	9	22.5	7.5	6

(a) Attributes of the films which the user doesn't watch before

	Mystery	Sci-Fi	Drama	Thriller	Horror	Suka
Angel & Demons	22.5			7.5		30
The Ring	22.5				6	28.5
Arrival	22.5	9	22.5			54

User A	22.5	9	22.5	7.5	6

(b) Calculation and recommendation

Figure 6. The system calculates suka scores to the films which the user doesn't watch before and recommends the film to the user

V. MODEL EVALUATION

A. Qualitative Evaluation

With the calculated user value, we can make predictions on films that the user has never watched before. This could be done by multiplying the user value with each attribute owned by the film as shown in Fig. 7. The sum of all values will be added together and the film with highest value will be recommended to the user like described above and in Fig. 6.

In practice, the authors have tested the models that have been made with films, where the films have been watched by previous users. The test results show that the predicted scores are quite close to the users' ratings that can be seen in Fig. 8. From this result, we could think that the model created based on user value has a good level of accuracy.

Comedy	3.407407
Drama	3.335821
Action	3.412698
Romance	3.247863
Adventure	3.259259
Thriller	2.892857
Crime	3.583333
Fantasy	3.184932
Children	3.206349
Animation	3.105263
Sci-Fi	2.860000
Horror	2.083333
Mystery	2.842857
IMAX	2.933333
Musical	3.303571
Documentary	4.312500
War	4.100000
Western	3.375000

Figure 7. Multiply features weight by user rating to get user's value

imdbId	userRating	value
1790809	3.5	4.9
3896198	4.0	4.7
4972582	0.5	1.5
3401882	4.5	3.4
4116284	3.5	4.9
4425200	4.0	4.9
5462602	4.0	4.4
3874544	4.5	4.8
6184894	4.0	3.4
6438918	4.5	3.4
6648926	4.0	3.4
493405	4.0	5.0
1412528	3.0	4.4
1469304	4.0	4.5
2334871	4.0	4.5

Figure 8. Comparison between user rating and value prediction

B. Common Issues Evaluation

The use of content-based filtering methods has several obstacles that may cause a lot of inaccuracies. Therefore, some changes are needed to overcome it.

Firstly, humans tend to have feelings that are easy to change. This also applies to the movie that person consumed. The existing model cannot recognize this change quickly due to the existing user value. One way to overcome this is to provide a feature to re-add new recommendations in other words to recalculate the user value without removing the previous user value.

Secondly, the proposed system will have difficulty giving film recommendations to the new users. This is due to the lack of user data that does not even exist yet. Though the thing determining the long-term use by the user is the experience of a new guy using the system at the beginning. To overcome this problem, we can use the CF method. This means that we could provide movie recommendations that have been previously recommended to others based on the film that the user likes in the beginning. When the data is sufficient to do the CB method, the proposed system will automatically switch CF method to CB method, and provide recommendations based on user value.

Figure 9 shows an example with comparison between two films for a user and its recommendation.

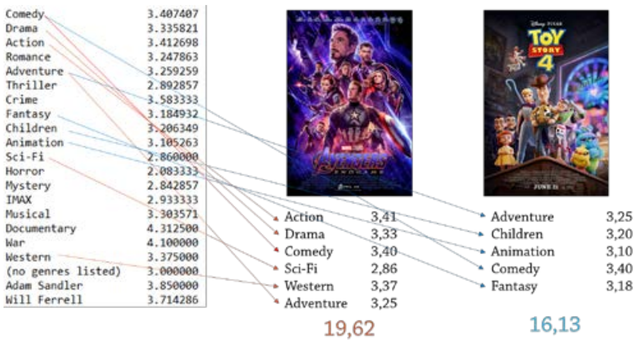

Figure 9. An example for the proposed movie recommender system

VI. CONCLUSION

In this article, a movie recommender system was built by the content-based filtering method. The dataset was obtained from MovieLens and added several attributes from OMDb API to increase the accuracy. The core of this proposed system is to obtain user value which will be used to calculate the presentation of user interests in films that have not been watched before.

REFERENCES

[1] IGI Global, Disseminator of Knowledge, website: https://www.igi-global.com/dictionary/recommender-system/24730, accessed on Aug. 20, 2019.

[2] Rajesh Khadka, Machine Learning Types #2, website: https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1, accessed on Aug. 20, 2019.

[3] C.-S. Mike Wu, D. Garg, and U. Bhandary, "Movie recommendation system using collaborative filtering," Proc. IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Nov. 2018, pp. 11-15, Beijing, China, doi: 10.1109/ICSESS.2018.8663822.

[4] B. R. Cami, H. Hassanpour, and H. Mashayekhi, "A content-based movie recommender system based on temporal user preferences," Proc. 3rd Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Dec. 2017, pp. 121-125, Shahrood, Iran, doi: 10.1109/ICSPIS.2017.8311601.

[5] N. Yi, C. Li, X. Feng, and M. Shi, "Design and implementation of movie recommender system based on graph database," Proc. 14th Web Information Systems and Applications

Conference (WISA), Nov. 2017, pp. 132-135, Liuzhou, China, doi: 10.1109/WISA.2017.34.

[6] N. Immaneni, I. Padmanaban, B. Ramasubramanian, and R. Sridhar, "A meta-level hybridization approach to personalized movie recommendation," Proc. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Sep. 2017, pp. 2193-2200, Udupi, India, doi: 10.1109/ICACCI.2017.8126171.

[7] H. W. Chen, Y. L. Wu, M. K. Hor, and C. Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," Proc. 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Jul. 2017, pp. 504-509, Ningbo, China, doi: 10.1109/ICMLC .2017.8108968.

[8] K. Verstrepen and B. Goethals, "Unifying nearest neighbors collaborative filtering," Proc. the 8th ACM Conference on Recommender systems, Oct. 2014, pp. 177-184, California, USA, doi: 10.1145/2645710.2645731.

[9] MovieLens, website: https://movielens.org/, accessed on Aug. 20, 2019.

[10] IMDb, website: https://m.imdb.com, accessed on Aug. 20, 2019.

[11] OMDb API, website: http://www.omdbapi.com, accessed on Aug. 20, 2019.