ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 10 Issue 12 December 2023

ART SMART: An Enhanced Content-Based Filtering Algorithm using Levenshtein Distance for Art Recommendation System

[1] Alain Jared N. Buot, [2] Kyle-Jie B. Dizon, [3] Mark Christopher R. Blanco, [4] Vivien A. Agustin, [5] Raymund M. Dioses

[1] [2] [3] [4] [5] Computer Science Department, Pamantasan ng Lungsod ng Maynila, Manila Philippines
[1] ajnbuot2019@plm.edu.ph, [2] kjbdizon@plm.edu.ph, [3] mcrblanco@plm.edu.ph, [4] vaagustin@plm.edu.ph,
[5] rmdioses@plm.edu.ph

Abstract*— This paper attempts to modify the Content-Based Filtering Algorithm (which is one of the known algorithms in Recommender Systems) using the Levenshtein Distance for Art Recommendation System. Recommender Systems are vital in today's age since there is so much information available on the Internet, and these systems are the ones in charge of filtering these massive amounts of data to fit your interests. This paper focuses on one drawback of the algorithm which is "Overspecialization", that is when the algorithm recommends items to the user that are very much similar to the user's previous activities. The Researchers gathered the data from data.world which consists of different information about each artwork and its artist. The findings imply that the use of the modified algorithm has improved in comparison to the original. Just like the original Content-Based Filtering, it suggests to users artworks that are based on their previous interests, but it also recommends fresh and familiar types of artworks that may expand the user's interests.

***Index Terms*— Content-Based Filtering, Levenshtein Distance, Overspecialization, Recommender System.**

I. INTRODUCTION

Lung The internet is now more than a tool to humans, some can argue that their life depends on it. With the size of information that the internet can offer, users may become overwhelmed by it. This is where recommendation systems come in, these systems are made specifically to recommend users items that may be of interest to them [1]. Content-based filtering (CBF) is one of the many recommender systems, CBF has been a widely used recommender system in multiple websites such as e-commerce, video sharing websites and many more. Content-based recommendation systems seek to recommend items that are similar to what the user positively rated or searched for in the past. CBF method is built on information retrieval, analysis, and filtering [2]. Traditional content-based filtering generally uses text and classification techniques for creating user profiles and also for the attributes of the items [3].In fact, the core function of a content-based recommender is to find new, interesting items to recommend to the user by matching the attributes of a user profile, which stores preferences and interests, with the attributes of a content object or item [4].

Now that the world is in the information age or the age of the internet, multiple websites have been exposed according to the needs of humans. Examples are e-commerce and online store websites. Over the course of the pandemic, there is evident trend towards online shopping, and retailers amongst millennials [5]. Also, the adoption of digital art sales and exhibition channels, along with the rapid advancement in technology, have helped increase both the average number of times a piece of art is viewed and the audience for art purchasers more specifically in paintings, traditional and digital [6].

A. The Problem Statement

CBF has several drawbacks. It cannot generate great suggestions if an item does not contain the proper descriptions for categorization. Plus, with enormous amounts of data, scalability and sparsity is also a possible challenge it may face [7].

1. Recommending the same types of items. There are specific limitations for CBF that the researchers are going to tackle in this study, and it is its overspecialization problem (also known as lack of serendipity) where it advocates only the same types of items to the users thus not being able to recommend unexpected, yet suitable items [8].
2. The recommendations lacks diversity. The model's recommendations are solely derived from the user's current interests or what the user "liked", indicating that it has a restricted capacity to broaden the user's existing interests [15].
3. Algorithm often shows bias for top rated content only. Many recommendation algorithms reinforce popularity bias thus frequently recommending popular items thus making new items or less popular items a "cold start" problem [9].

B. Objective of the Study

1. To implement Levenshtein Distance to CBFA to help provide more various recommendations based on the user's profile

One objective of this study is to develop an enhanced Content-Based Filtering Algorithm (CBFA) that can provide recommendations with greater variance. The proposed algorithm is meant to offer users a chance to explore more new items while ensuring a certain level of relevance to their existing interests.

2. To provide two kinds of recommendations

The modified algorithm would allow the system to provide two kinds of recommendations: based on what the user recently searched and positively rated and based on what is the most searched and positively rated. To expand the diversity of the recommendations, and to avoid focusing on one specific type.

3. To give all items equal chance to be recommended

To make sure that there is an equal opportunity for all items to be recommended, regardless of their rating, a randomization process will be implemented. Which means that the items/artwork that will be up for recommendation will be carried out in a manner where every item, not focusing on its rating or perceived quality, will have an equal chance of being chosen.

II. RELATED WORKS

A. Recommender Systems

In today's age, the amount of information found in the internet is rapidly multiplying and becoming more complex but with the help of recommender systems the end users will not suffer from information overload [10]. Recommender systems are utilized in several platforms such as social media, e-commerce, video on demand websites and many more. This enables users to have easy access to contents that are much more suitable to their preferences [11]. In this study, focusing on Content-Based Filtering (CBF), how it functions is that it extracts the characteristic of an item in the database and compares it to the characteristics stored in a user profile [12].

B. Content-Based Filtering Problem

Content-Based Filtering is unable to generate serendipitous suggestions, meaning that CBF lacks the ability to suggest items that users do not expect to see but still somewhat familiar [13]. A study states that users could find recommender systems more useful if it is able to recommend items that are new or unexpected. This is currently the disadvantage of CBF since it tends to overspecialize the item-selection and only the very similar items of the previous items consumed by the user are recommended [14].

If the recommender system is solely focused on recommending a certain set of items (overspecialization), lack of also diversity arises. In which all of the recommended items are too similar to one another, because these are predicted to be "liked" by the user. Thus, overemphasizing accuracy in recommendation systems can lead to a lack of freshness or diversity, as it may result in a limitation on the variety of recommended items, ultimately making the recommendations excessively predictable [15].

One more problem that CBF faces is, many recommendation algorithms reinforce the popularity bias in rating data by frequently recommending popular items while not giving enough exposure to less popular ones [9].

C. Levenshtein Distance

Levenshtein presented the method in 1966 where it is the earliest known to use a distance function that is appropriate in the presence of insertion and deletion errors for sequence comparisons. The Levenshtein Distance Algorithm is commonly used to determine or measure the distance similarity of sets of strings. The lower the distance the higher the similarity [16]. Levenshtein Distance is a string comparison technique that counts the number of edit operations needed to make one string the same as another. For example, the Levenshtein distance between the Massau and Tongan cognate words tolu is 0, and the difference of tolu and Javanese telu is 1. It is said that this is normalized by dividing the length of the longest word, so the distance between tolu/telu is 0.25 [17].

III. METHODOLOGY

A. The Original CBFA Model

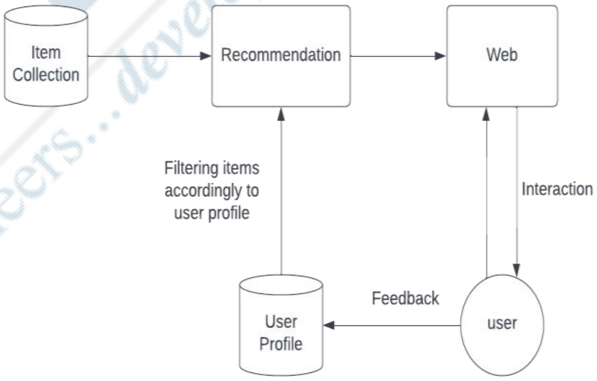

Figure 1. Content-Based Filtering for Recommendation, model by Robin van Meteren (2000), explaining how the CBFA works in recommending items based on user's feedback.

In figure 1. Shows how content-based filtering works when recommending and this model has been proposed by Robin Van Meteren [18]. It simply shows the process of how CBFA works, the first step is having an "item collection" where items and its descriptions are stored which allows identification for each item. Then, user interaction through the "Web" interface. Users have the ability to rate, search or give feedback to specific items, through those actions, the user's preference will be indicated. These interactions or actions would be stored in the specific user's "user profile" serving it as reference for the system. Next, is the "Recommendation" phase where the system suggests items to the user.

B. Conceptual Framework

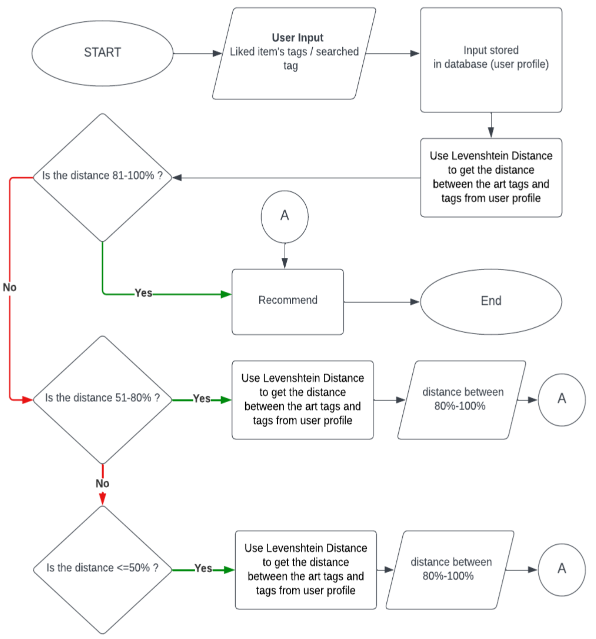
START
User Input
Liked item's tags / searched tag
Input stored in database (user profile)
Use Levenshtein Distance to get the distance between the art tags and tags from user profile
Is the distance 81-100% ?
A
No
Yes
Recommend
End
Is the distance 51-80% ?
Yes
Use Levenshtein Distance to get the distance between the art tags and tags from user profile
distance between 80%-100%
A
No
Is the distance <=50% ?
Yes
Use Levenshtein Distance to get the distance between the art tags and tags from user profile
distance between 80%-100%
A

Figure 2. Conceptual Framework of Enhanced CBFA.

Figure 2 shows the conceptual framework of the proposed modification of CBFA using Levenshtein Distance. At the start of the algorithm, the user input is defined to be what tags that the user “liked” and “searched” , then it will be stored in a specific user profile in the database. The next step is where the Levenshtein Distance is applied, instead of the conventional Cosine Similarity that is used in most recommender systems, the researchers used Levenshtein Distance to determine the distance between tags from the user profile and item tags. Thus, when the distance between the tags is between 81% to 100% it would be included in the list to be recommended right away. While when the distance is between 51% to 80%, it will undergo another Levenshtein Distance calculation wherein it's 100% and 80% similarity will be recommended. The same case shall happen at 50%. Basically, in this way the system not only recommends what is completely identical to the item that the user interacts with, but also it recommends what’s similar and slightly similar to the item in order to give the user a variety of new but familiar items to see.

There are two types of recommendation results in this modified system, first is via most searched / most liked (“Just For You”) and recently searched / recently liked (“Recent Interests”). Therefore this system will provide two sets of recommendations for the user.

C. Levenshtein Distance Equation

$$
\begin{aligned}
lev_{a,b}(i,j) = \{(i,j)\ \ min\{lev_{a,b}&(i-1,j) \\
&+ 1\ lev_{a,b}(i,j-1) \\
&+ 1\ lev_{a,b}(i-1,j-1) \\
&+ 1_{(a_i \neq b_j)}\ \ if(i,j)\ = 0 \quad \text{otherwise}
\end{aligned}
$$

where,
a = string#1
b = string#2
i = the terminal character position of string #1
j = the terminal character position of string #2

The Levenshtein Distance equation is a mathematical formula used to calculate the difference between two strings. It measures the minimum number of single-character edits required to transform one string into another. The equation considers three possible editing operations: insertion, deletion and substitution.

To calculate the distance between two strings, the algorithm compares each character of the first string with each character of the second string. It examines the differences between corresponding characters and determines the edit operation needed to make them a match.

The equation iterates over the entire length of both strings, considering all possible combinations of insertions, deletions, and substitutions. The goal is to find the edit sequence that yields the minimum number of operations required to transform one string to another.

D. Materials utilized for the system

For the development of the ArtSmart system, the researchers chose the Python programming language due to its flexibility and wide range of libraries. One of the libraries that is mainly used is the Tkinter library, this is used for the GUI or Graphical User Interface, it enables the creation of interactive and visually appealing components. The database used for the system is MySQL in order to store and retrieve crucial information that will be used in the system such as user profiles, user-related data, dataset details and dataset-related information. With all these tools, the system is able to ensure efficient storage, retrieval and manipulation of data, completely enhancing its overall functionality and performance of the Artsmart recommender system.

E. Data Pre-processing

a) Dataset

The dataset used for this study is from *data.world*. “Artworks” is the name of the dataset and is provided by MoMA or Museum of Modern Art. The dataset consists of approximately 132,000 rows of artworks with 25 columns each having information about the Artwork itself and its Artist.

To make the dataset clean and suitable for the system, unnecessary columns, blank fields, and duplicate rows were removed. The only columns that will be used are: "Artwork title," "Artist name," and "Medium," as they serve as descriptive tags for the artworks.

b) User Input Retrieval

Art Smart uses a register and login function to provide unique recommendations to each registered user. The system gathers information from the user via the search bar and “Like” buttons. The search bar located on the top is where users can enter words, it can be an artist, an artwork, or a medium. If the system detects that the input word is a medium or tag, it will store it in the user's profile. Now the second way is the "Like" buttons located on the right side of each artwork and whenever a user clicks this, it will store all its associated tags to the user profile.

F. Tag Extraction

Each user has two profiles, one is for the recently liked and recently searched tags, and the other one is for the most liked and most searched tags. The system then counts each of the user's profile and gets the top three tags and uses this as the basis for the filtering.

G. Enhancement of CBF using Levenshtein Distance

The algorithm starts with initializing the variables such as 'X' for storing the lowest edit distances, then 'Y' as the divisor to calculate the averages, and 'Ave' for the total average of all distances. There are also multiple lists in this algorithm for storing the different items that will be recommended to the user. The naming of the variable has two parts which have meanings, the first is the indicator that tells the proportion of items being categorized relative to the total number of items, the next is on what category the item is being put into based on its similarity percentage. The next step is retrieving the user's top stored tags and the algorithm iterates through each item and its associated tags. The distance between the tag of an item and the tag from the user profile will be calculated using the Levenshtein Distance. The Levenshtein Distance measures the minimum edit distance or the number of operations (insertion, deletion, substitution) needed to make one string (in this case, the item tag) identical to the other (tag from user profile). Thus, the algorithm keeps track of the lowest distance for each item tag

By summing up all the lowest distances ('X') and dividing it by a divisor ('Y') which is just the number of tags an item has, the average distance ('Ave') is calculated, and then the average is converted into a percentage. Depending on the percentage distance, the algorithm shall assign the item to the corresponding category; "onehundred_love", "onehundred_like", or "onehundred_dislike". The "love" signifies that the item is highly similar (81%-100%), the "like" is moderately similar (51%-80%), while the "dislike" is only slightly similar (50% below).

The items in "onehundred_love" are recommended on the top right away. The items in the "onehundred_like" and "onehundred_dislike" are adjusted to multiply by 1.25 and 2 respectively. This adjustment is necessary to allow the system to get the highly and moderately similar items of each category. And lastly, based on the updated percentages, the algorithm will assign items to the remaining categories; "eighty_love," "eighty_like," "fifty_love," "fifty_like," etc. Then the algorithm recommends items from the "onehundred_love," "eighty_love," "eighty_like," "fifty_love," and "fifty_like" categories to the user.

IV. RESULTS AND DISCUSSION

A. Problem 1

Table 1. Item Profile, User Profile and the result of the Levenshtein Distance

Artwork Title: Untitled 1962 Artwork Tags: "sand" , "synthetic polymer" , "canvas" User's Top Tags: "oil" , "ink" , "canvas"
Levenshtein Distance of sand and oil: 4 Levenshtein Distance of sand and ink: 3 Levenshtein Distance of sand and canvas: 4
Levenshtein Distance of synthetic polymer and oil: 15 Levenshtein Distance of synthetic polymer and ink: 16 Levenshtein Distance of synthetic polymer and canvas: 16
Levenshtein Distance of canvas and oil: 6 Levenshtein Distance of canvas and ink: 5 Levenshtein Distance of canvas and canvas: 0
Average Levenshtein Distance: 6.0 or 40%

The table above is a sample of how the system calculates the average Levenshtein Distance. In each line, the first tag is from the item, while the second tag is from the user profile.

Using the formula of Levenshtein Distance, each item tag is compared to the user profile's top three tags individually. The system will gather the lowest distances of each item tag, get the sum of all the lowest distances, and divide it by the number of item tags or medium. The average distance will then be converted to percentage so that the system can use this on how similar an item is to a user profile.

a) CBF using Levenshtein Distance

Table 2. Ten sample rows of artworks

	Artwork Title	Medium or Tags	%
1	Untitled 1962	sand, synthetic polymer, canvas	40%
2	Ukulele	oil, charcoal, paper, canvas	75%
3	Still Life	oil, bronze, plywood	64%
4	"M'Amenez -y"	oil, enamel, cardboard	64%
5	Green Lush Forest	dyed cotton, muslin	25%
6	Murder in the jungle	oil, composition board	30%

7	Poltergeist	synthetic polymer, canvas, wood	40%
8	The flame and the diver	oil, canvas	100%
9	Number 5-58	synthetic polymer, canvas	25%
10	Homestead	tempera, oil, composition board	30%

Table 1 is a sample of 10 rows of data from the dataset. The first column is the title of the artwork, the second is the medium of the artwork or its descriptive tags, lastly is the average Levenshtein Distance of each row. The User's Top Tags for the simulation seen in Table 1 are "oil", "ink", and "canvas", which is the same as the one seen in the image in part *A* in chapter IV.

Artwork 8 is the only item that is highly similar therefore, it is the only item to be put in the "onehundred_love" category. Artworks 2 - 4 are moderately similar so the system will adjust their distances by multiplying it to 1.25. Then, the items that now have high similarity will be the second to be recommended, and next to this is the moderately similar items after the adjustments. Now for the rest of the rows, it will also be adjusted by multiplying it by 2. The items that have high similarity after the adjustments will be the next to be recommended, the items that have moderate similarity after the adjustments will be the last to be recommended, and the items that still have low similarity will not be recommended anymore. After this process, this will be the recommendation.

Table 3. Recommendation of modified CBF

	Artwork Title	Medium or Tags	%
8	The flame and the diver	oil, canvas	100%
2	Ukulele	oil, charcoal, paper, canvas	93.75%
3	Still Life	oil, bronze, plywood	80%
4	"M'Amenez-y"	oil, enamel, cardboard	80%
1	Untitled 1962	sand, synthetic polymer, canvas	80%
7	Poltergeist	synthetic polymer, canvas, wood	80%
6	Murder in the jungle	oil, composition board	60%
10	Homestead	tempera, oil, composition board	60%

B. Problem 2

Applying the process of the enhanced algorithm, the system has implemented 2 kinds of recommendations to gain more diverse recommendations that are based on the user's profile. The user profile contains two types that will be used for the recommendation process.

Table 4. The user profile's two types of tags

User's Top Tags: "oil", "ink", "canvas"
User's Recent Tags: "synthetic polymer", "canvas", "acrylic"

Table 5. Diversified Recommendations

Based on User's Top		Based on User's Recent	
Artwork Title	%	Artwork Title	%
The flame and the diver	100%	Number 5-58	100%
Ukulele	93.75 %	Untitled 1962	87%
Still Life	80%	Murder in the jungle	100%

The result presented on table 5 shows the top three items of each recommendation. The first is based on the user's top interacted tags and the second is the recently interacted tags.

C. Problem 3

In the system, users have the option to rate an item and when it is rated, other users can see it. This rating will not affect its likeliness of being recommended to a user. Each part of the recommendation list is randomized so, an item with 1 star will have an equal chance to being recommended as an item with 5 stars. For example, using the results in table 3. the recommendation will now look something like this.

Table 6. Recommendation after randomization

	Artwork Title	Medium or Tags	Rating
8	The flame and the diver	oil, canvas	3.9★
2	Ukulele	oil, charcoal, paper, canvas	2.2★
4	"M'Amenez-y "	oil, enamel, cardboard	4.5★
3	Still Life	oil, bronze, plywood	2.9★
7	Poltergeist	synthetic polymer, canvas, wood	4.7★
1	Untitled 1962	sand, synthetic polymer, canvas	2.8★
6	Murder in the jungle	oil, composition board	3.2★
10	Homestead	tempera, oil, composition board	4.8★

V. CONCLUSION

Artsmart's algorithm is a modification of the Content-Based Filtering Algorithm (CBFA), tailored to resolve some of the issues that CBFA encounters. Specifically, the algorithm aims to recommend items that are not overspecialized or repetitive. To achieve this goal, the researchers incorporated Levenshtein Distance to compare and find similarities among tags. This modification enables the algorithm to recommend items that are both similar and slightly different from the original item, providing users with a variety of new, yet familiar items to explore. The system also provides two types of recommendations based on the most interacted tags and recently interacted tags, respectively, to broaden the range of recommended items. Additionally, the algorithm randomizes the items to recommend, giving all items an equal opportunity to be recommended regardless of the user's rating.

In conclusion, the modification of the CBFA using Levenshtein Distance has successfully addressed the issues presented in the problem statement. The recommended items displayed by the system have proven to be relevant and similar to the user's interests based on their profile, while also suggesting new and somewhat similar tags that may spark the user's interest. Overall, Art Smart's algorithm has effectively resolved the issues of overspecialization and repetitiveness in recommending items to users.

REFERENCES

[1] Geetha, G., Safa, M., Fancy, C., & Saranya, D. (2018). A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System. Journal of Physics: Conference Series, 1000, 012101. https://doi.org/10.1088/1742-6596/1000/1/012101

[2] Stitini, O., Kaloun, S., & Bencharef, O. (2022). An Improved Recommender System Solution to Mitigate the Over-Specialization Problem Using Genetic Algorithms. Electronics, 11(2), 242. https://doi.org/10.3390/electronics11020242

[3] Shoval, P., Maidel, V., & Shapira, B. (2008). AN ONTOLOGY-CONTENT-BASED FILTERING METHOD. International Journal "Information Theories & Applications, 15. http://sci-gems.math.bas.bg:8080/jspui/bitstream/10525/88/1/ijita15-4-p01.pdf

[4] Lops, P., de Gemmis, M., & Semeraro, G. (2010). Content-based Recommender Systems: State of the Art and Trends. Recommender Systems Handbook, 73–105. https://doi.org/10.1007/978-0-387-85820-3_3

[5] Akram, U., Fülöp, M. T., Tiron-Tudor, A., Topor, D. I., & Căpușneanu, S. (2021). Impact of Digitalization on Customers’ Well-Being in the Pandemic Period: Challenges and Opportunities for the Retail Industry. International Journal of Environmental Research and Public Health, 18(14), 7533. https://doi.org/10.3390/ijerph18147533

[6] Sidorova, E. (2022). Global Art Market in the Aftermath of COVID-19. Arts, 11(5), 93. https://doi.org/10.3390/arts11050093

[7] Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. Expert Systems with Applications, 89, 404–412. https://doi.org/10.1016/j.eswa.2017.08.008

[8] De Gemmis, M., Lops, P., Semeraro, G., & Musto, C. (2015). An investigation on the serendipity problem in recommender systems. Information Processing and Management, 51(5), 695–717. https://doi.org/10.1016/j.ipm.2015.06.008

[9] Abdollahpouri, H. (2019, July 31). The Unfairness of Popularity Bias in Recommendation. arXiv.org. https://arxiv.org/abs/1907.13286

[10] Afoudi, Y., Lazaar, M., & Al Achhab, M. (2021). Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. Simulation Modelling Practice and Theory, 102375. https://doi.org/10.1016/j.simpat.2021.102375

[11] Bourgais, M., Zanni-Merk, C., Fatali, R., & Alizada, N. (2022). Avoiding the Overspecialization of Recommender Systems in Tourism with Semantic Trajectories, Initial Thoughts. Procedia Computer Science, 207, 1933–1942. https://doi.org/10.1016/j.procs.2022.09.252

[12] Cortez, D. M. A., Cordero, N. J. J., Canlas, J. C., Mata, K. E., Regala, R. C., Blanco, M. C. R., & Alipio, A. J. (2022). Modified Content-Based Filtering Method Using K-Nearest Neighbors and Percentile Concept. Modified Content-Based Filtering Method Using K-Nearest Neighbors and Percentile Concept, 100(1), 14–14. https://ijrp.org/paper-detail/3090

[13] Tewari, A. S., Singh, J. P., & Barman, A. G. (2018). Generating Top-N Items Recommendation Set Using Collaborative, Content Based Filtering and Rating Variance. Procedia Computer Science, 132, 1678–1684. https://doi.org/10.1016/j.procs.2018.05.139

[14] Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo, J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences, 180(22), 4290–4311. https://doi.org/10.1016/j.ins.2010.07.024

[15] Karanam, M. B. (2010). Tackling the problems of diversity in recommender systems. https://krex.k-state.edu/handle/2097/6981

[16] Putera Utama Siahaan, A., Aryza, S., Hariyanto, E., Rusiadi, Hasudungan Lubis, A., Ikhwan, A., & Len Eh Kan, P. (2018). Combination of levenshtein distance and rabin-karp to improve the accuracy of document equivalence level. International Journal of Engineering & Technology, 7(2.27), 17. https://doi.org/10.14419/ijet.v7i2.27.12084

[17] Greenhill, S. J. (2011). Levenshtein Distances Fail to Identify Language Relationships Accurately. Computational Linguistics, 37(4), 689–698. https://doi.org/10.1162/coli_a_00073

[18] Robin Van Meteren, & Maarten Van Someren. (n.d.). Using Content-Based Filtering for Recommendation 1. http://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf