A Content-Based Collaborative Filtering Movie Recommendation System using Keywords Extractions

Mtuthuko Mngomezulu
School of Computer Science and Applied Mathematics
The University of the Witwatersrand
Johannesburg, South Africa
Email: 1422811@students.wits.ac.za

Ritesh Ajoodha
School of Computer Science and Applied Mathematics
The University of the Witwatersrand
Johannesburg, South Africa
Email: ritesh.ajoodha@wits.ac.za

***Abstract*—The main focus of this research is to develop a Content-Based Collaborative Filtering model that uses different automated keyword extraction techniques to recommend movies to a user. Recommender systems predict consumers' preferences for products and provide proactive suggestions for items they would enjoy. Collaborative filtering, content-based, and hybrid recommendation models are the most common types of recommendation models.Collaborative filtering generates suggestions based on previous interactions between the user and the item, whereas the majority of content-based recommendations are based on item comparisons. The majority of hybrid recommender systems are made up of a mix of collaborative filtering and content-based recommender models. The Content-Based method was used as the main model in this study, with Term Frequency - Inverse Document Frequency (TF-IDF) and Rapid Automatic Keyword Extraction (RAKE) algorithms serving as keyword extractors. A total of 244 movies were recommended using keywords from each extractor, with the highest average of 33% of the movies recommended from each being identical. Taking comparable movies into account, we can propose them to a user.**

***Index Terms*—Recommendation System, Content-Based, Collaborative Filtering,TF-IDF,RAKE, cosine similarity matrix**

I. Introduction

A recommendation system is a program that employs various filtering algorithms to deliver suggestions and forecasts to users. Primarily, demographic, content-based (CB), and collaborative filtering were used (CF) [1], recommendation systems have been continuously developing and changing, with the introduction and usage of many different models.Collaborative filtering is a mechanism for filtering or calculating predicted user preference based on users' previous interactions [4] [5], Content-based recommenders, on the other hand, propose related goods based on a specific item [1] .

The movie business has grown in recent years, with hundreds of films made practically every year; as a result, consumer over-choice has become a problem. As a result, both business and research have embraced movie recommender systems [1] [6] . Many approaches have been taken in developing recommendation systems [6] [9] [10] [12] . [9] discusses a clustering approach to collaborative filtering movie recommendation systems based on K-means clustering and K-Nearest Neighbor. [10] looks into an enhanced auto encoder recommendation system and its application in collaborative filtering movie recommendation by employing a deep learning technique and measuring the divergence of the projected ratings with Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

[6] investigates a totally content-based movie recommendation system with feature extraction utilizing a neural network to entirely recommend movies. Word2Vec CBOW is used to extract feature vectors and calculate the similarity of movies based on the extracted features. Keyword extraction has also been a useful tool and a significant contribution in the development of recommendation systems. [12] created a news subject recommendation system based on keyword extraction from internet news data using the Rapid Automatic Keyword Extraction (RAKE) algorithm. With a focus on semantic similarity of tweets and its influence on tag recommendation issues, [13] created a Micro-blogging Hash Tag recommendation system based on semantic TF-IDF by employing semantic similarity as a weighting schema along TF-IDF.

In this study, we offer a collaborative filtering recommendation system based on content. We utilize a keywords feature extraction technique for the content-based approach, using the Rapid Automatic Keyword Extraction (RAKE) and TF-IDF algorithms to the movie plot to extract keywords, which we then use to calculate the cosine similarity matrix and propose movies with similar plots. We propose a model that predicts user ratings using a weighted average method with cosine similarity as the weights for the collaborative-filtering technique. To assess the accuracy of the model predictions, we use the Root Mean Square Error as the error measure.

II. RELATED WORK

Since the publication of the first articles on collaborative filtering in the mid-1990s, recommendation systems have been an active area of research [14]. Recommender systems have primarily been utilized in recent years to forecast consumers' product preferences and make proactive choices for items they might appreciate. The most frequent forms of recommendation models [2] include collaborative filtering, content-based, and hybrid recommendation models(a combination of content-based and collaborative filtering), see Fig. 1 and 2 for a general overview.

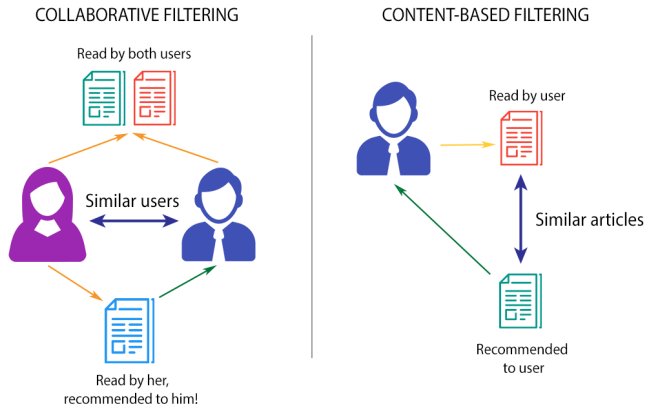

Fig. 1. General overview of CB and CF Rec Systems [15]

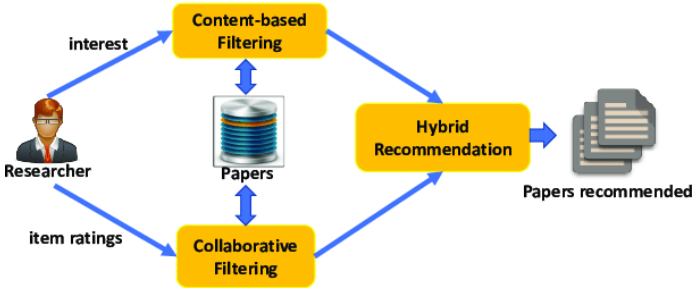

Fig. 2. General overview of a hybrid recommendation system based on researolch paper recommendation [16] .

A. Content-Based

These recommendation systems analyze user content to identify similarities and recommend content to users. The user interests profile is then created to assist the algorithm in identifying the appropriate items, [17]. CB systems necessitate proper ways for representing items and producing user profiles, as well as strategies for comparing the person profile to the item representation.

In contrast to CF recommendation systems, which typically employ a use-item (or user-user, item-item) matrix as input, this method uses some information that can extract item characteristics as training data [6]. Some content-based movie recommendation systems employ movie metadata (e.g. movie narrative, genre, cast, director of the movie) as input [19] [20] [6] and some use audio and visual features [21] [6] . Text documents are the other most common information source for content-based recommendation systems.

B. TF-IDF

To define the objects, a collection of descriptors or terms, often Term Frequency (TF) and Inverse Document Frequency (IDF), are utilized. [22]

Term frequency and inverse document frequency (TF-IDF) can identify key terms or phrases in papers [22] [23].

If a word appears infrequently but frequently in one or a few articles, it most likely plays an important part in the article. Furthermore, the greater the TF IDF of a word, the more essential it is in the article. The Term-Frequency IDF is determined by combining the Term-Frequency and the Inverse-Document-Frequency [22].

TF-IDF is the most commonly used weighting approach for describing documents in the vector space model [22] [24] [25] . By using semantic similarity as a weighting schema with TF-IDF, [13] constructed a Micro-blogging HashTag recommendation system based on semantic TF-IDF.

In [26], the authors exploit web data by developing a model-based recommendation system that adapts the bag of words model to cope with the RDF movie dataset while also expressing each attribute as a unique vector of weights derived by the TF-IDF approach.

C. RAKE

Rapid Automatic Keyword Extraction (RAKE) is an unsupervised algorithm for extracting keywords from research papers [27] [28]. Keywords in RAKE include several content words that are instructive rather than punctuation and stop words [27] [30].RAKE can be applicable to any sort of text, regardless of domain, especially ones that do not follow clear linguistic tradition, and the input parameters for RAKE are a list of stop words, phrase delimiters [27] [28].According to [12], the performance of RAKE's processing time is slower than that of most keyword extraction methods in parallel with larger data sets, making RAKE suitable for processing vast amounts of data.

Rapid Automatic Keyword Extraction, like TF-IDF, has been a valuable technique in the creation of recommendation systems. The authors of [12] employed RAKE on internet news data to aid in the development of a news subject recommendation engine. The authors of [29] suggested an auto-tagging approach that includes automated keyword extraction and tagging, as well as Rapid Automatic Keyword Extraction for pre-processing and keyword extraction.

D. Collaborative Filtering

Collaborative filtering creates suggestions by learning from prior user/item interactions, either explicitly (e.g., previous user ratings) or implicitly (e.g., user browsing history) [2]. As data to work with, collaborative recommender systems rely mostly on overlapping data, such as user ratings or user use

history. Unlike CB, CF is offered as one of the deep learning recommendation systems that outperforms the others [18].
The authors in [9] explain and suggest a clustering technique to collaborative filtering movie recommendation systems, which is based on K-means clustering and the K-Nearest Neighbor algorithm.
The authors of [10] investigate an improved auto-encoder recommendation system and its application in collaborative filtering movie recommendation by employing a deep learning technique and measuring the divergence of the projected ratings with Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

III. Methodology

A. Data

The dataset utilized in this paper is the MovieLens Datasets [11] [31]. We use two sets of MovieLens datasets, one retrieved from the MovieLens website, which has 27753444 ratings and 1108997 tag applications scattered across 58098 movies and was gathered between January 9, 1995 and September 26, 2018, and was made by 283228 people [31]. The second dataset is likewise part of the MovieLens dataset; it is known as the MovieLens 100K [11] and was obtained from the Kaggle website. The dataset contains 100,000 ratings (1-5) from 943 users on 1682 movies, with each user rating at least 20 movies [11] [32].

1) Data Extraction: For the CB part we extracted 45466 entries among the 58098 movie data from the MovieLens dataset. The data had 23 column feature entries, among those 23 we only extracted two features (Movie title and Movie Plot). And for the Collaborative Filtering part of the system we extracted 1682 extries from the 100K dataset, from among those extries we extracted 3 features to use (user id, user ratings, movie id).

We use the MovieLens datasets because they contain a large amount of data about 58098 movies [11] [32] and their movie plots, making it a good match for our content-based section of our recommender system, which seeks to recommend movies based on the movie plot, and the movie plots are properly detailed, making them a good choice for using keyword extractors from them. Furthermore, the data set contains a significant number of movie ratings from a big number of individuals, with each user rating at least 20 movies which we an use to predict user ratings using the collaborative filtering section of our recommendation system.

B. Models and Feature Extraction

We define two types of recommendation system models in this paper: content-based (CB) and collaborative filtering (CF). For the CB, we utilize two sets of keyword extractors for feature extraction: TF-IDF (from the Scikit-learn feature extractor library) and RAKE (from the rake natural language toolkit library) on a set of 45466 movie plots. Each keyword extractor can be utilized independently of the others.

The CF portion of the recommender system uses 1682 entries of user ratings from the 100K Movielens dataset and uses the weighted average and cosine similarity value methods of user movie ratings to calculate and predict user ratings. The MSE in this study is used to evaluate the accuracy of the evaluation predicted by the model. Two models, SVD and KNN, are used to improve RMS error assessment. We also employ NMF,SVD++, KNN with Z Score, and CoClustering, models to compare the performance of our SVD and KNN

In all techniques of our recommender system, we employ cosine similarity from the scikit-learn library. In our content-based model, we use it to compute the similarity between two movies based on their movie plot, and in our collaborative filtering model, we use it to determine the similarity between user ratings and movie title.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Fig. 3. Cosine similarity is calculated using this formula

C. System Overview

The structure of the CB recommendation system used is shown in Fig. 5

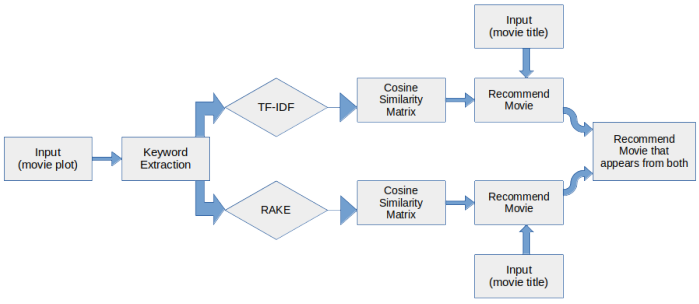

Fig. 4. Structure of CB model used.

D. Evaluation

To evaluate our CB recommendation system, we compare the results from the two keyword extractors and divide the total number of identical movies recommended by the two models by the total number of movies recommended.

IV. RESULTS

In this section, we will go through the outcomes of our recommendation system. A total of seven movie titles were chosen to recommend 20 films with comparable plots.Using cosine similarity matrices generated by keyword extractors TF-IDF and RAKE on movie plots, we used each movie title to recommend movies with similar movie plots.

```
[11]: 9                        GoldenEye
      10          The American President
      11     Dracula: Dead and Loving It
      12                           Balto
      13                           Nixon
      14                 Cutthroat Island
      15                          Casino
      Name: title, dtype: object
```

Fig. 5. Chosen movie titles

From the set, a total of 244 movies were recommended, with an average of 34 movies recommended per movie title put into the recommendation algorithm. We receive an average of 7 movies recommended per title supplied from the entire movies that were recommended,these are the movies that match from both keyword extractors, demonstrating that we can combine RAKE and TF-IDF to recommend movies using movie plot and extracting the similar movies acquired by utilizing the keywords extraction models.

Table I shows the number of movies that were were the same from both the TF-IDF and RAKE 20 movie recommendations, where TF-IDF MisMatch shows the number of movies from the TF-IDF recommendation that were not the same as the other movies from the RAKE recommendation, and the same goes for the RAKE MisMatch column.

With matching suggestions for Casino being our highest combined matched recommendations, we propose 10 movies to viewers who want to see a film with a narrative similar to Casino. And The American President has the lowest combined recommendation, with only one film recommended. Because TF-IDF and RAKE utilize distinct ways to extract key phrases, the significant percentage of movies that deviate from both suggestions is expected.

The combined findings demonstrate that there is a significant chance that utilizing separate keyword extractors and picking movies that are the same from each extraction recommendation system would likely propose movies that the viewer will appreciate.

The test run on the recommendation system's Collaborative Filtering component to forecast user rating yielded a mean squared error of 1.0172812824757378 on the test set. To decrease this score, we first used KNN and SVD models, as well as 10 fold cross validation with RMSE and MAE as metrics.

Both training and testing times were quantified in terms of performance. When it came to training time, the KNN model outperformed the SVD, and when it came to testing time, the SVD outperformed the KNN model.

TABLE I
MATCHED AND MISMATCHED RECOMMENDATIONS.

Movie Title	**Movies Recommended**		
	TF-IDF & RAKE Match	***TF-IDF MisMatch***	***RAKE MisMatch***
GoldenEye	7	13	13
The American President	1	24	24
Dracula: Dead and Loving It	7	13	13
Balto	8	12	12
Nixon	6	14	14
Cutthroat Island	7	13	13
Casino	10	10	10

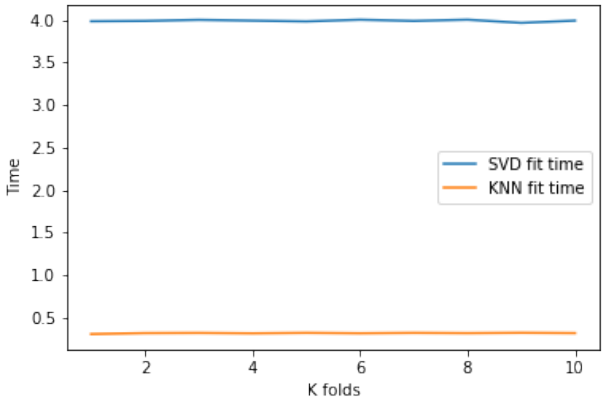

Fig. 6. SVD VS KNN with respect to training time

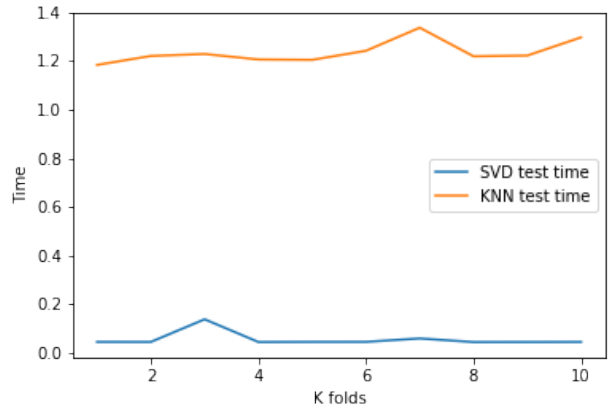

Fig. 7. SVD VS KNN with respect to testing time

The RMS error for the KNN model was 0.97, whereas the MAE averaged 0.77. The RMS error for the SVD model is typically 0.92, with an average MAE of 0.76.

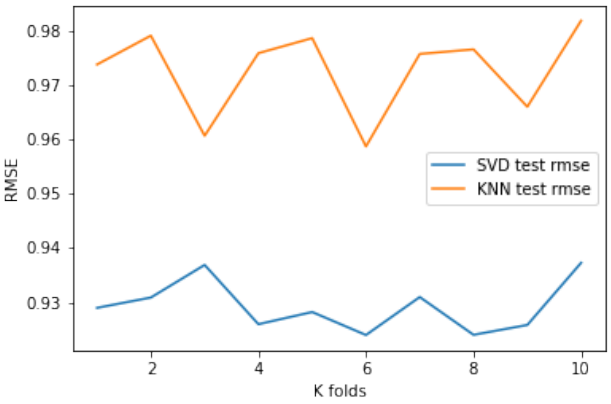

Fig. 8. RMSE

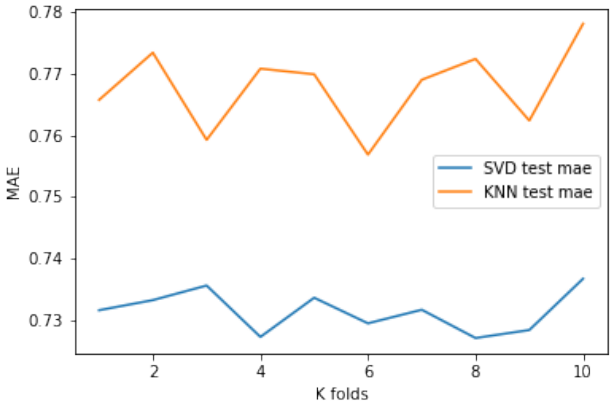

Fig. 9. MAE

To assess the validity of our KNN and SVD models, we performed an RMSE-MSE comparison test with four additional models: NMF,SVD++, KNN with Z Score, and CoClustering. As a baseline, we predicted the identical user's ratings using all of the models, and the SVD++ model performed better than all of the other models, with the lowest RMS error score of 0.90 and MA error value of 0.70. With 0.91 for the RMSE and 0.72 for the MAE, our SVD model has the second best per error scores. When compared to the other models, our KNN model performed the poorest, with error values of 0.95 for RMSE and 0.75 for MAE.

In general, SVD delivers more accurate predictions than KNN, but the findings also reveal that CoClustering model outperforms our basic KNN model, but KNN with Z Score outperforms both basic KNN and CoClustering model.

These findings demonstrate that the best model to use for predicting user ratings is the SVD++, and KNN with Z Score would outperform a standard KNN model when it comes to predicting user ratings.

Table II gives a detailed view of the results obtained.

TABLE II
RMSE - MAE

Model	RSME	MAE
NMF	0.941442	0.742425
SVD	0.918435	0.723637
SVDpp	0.904880	0.709138
KNNBasic	0.951915	0.751608
KNNWithZScore	0.933205	0.733548
CoClustering	0.939218	0.738580

ACKNOWLEDGEMENT

This work was supported by the Astron Energy Bursary. Computations were performed using High Performance Computing infrastructure provided by the Mathematical Sciences Support unit at the University of the Witwatersrand. This work is based on the research supported in part by the National Research Foundation of South Africa (Grant numbers: 121835).

CONCLUSION

WIth an avarage of 7 movies and 10 out of 20 movies being the highest number of movies being the same from the RAKE and TF-IDF systems. The results of combining TF-IDF and RAKE to suggest movies based on the movie plot demonstrate that we can combine RAKE and TF-IDF to recommend movies based on the movie plot and extract comparable movies obtained by using keywords extraction models. There is a significant chance that utilizing separate keyword extractors and picking movies that are the same from each extraction recommendation system would likely propose movies that the viewer will appreciate.

SVD++ is the best model to use for predicting user ratings, and KNN with Z Score would outperform a standard KNN model when it comes to predicting users' ratings. We can see another path to movie recommendation systems with the combination of a CF predicting user ratings with minimal error, which will help with the development of movie recommendation systems. For future work, we would like to explore other parts of recommendation systems, and research in detail the dangers that recommendation systems may impose on society.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Guti errez." Recom-mender systems survey.Knowledge-Based Systems", 46:109–132, 2013

[2] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning basedrecommender system: A survey and new perspectives. InACM Comput. Surv,volume 52, page 38, 2019

[3] Jun Shi Bo Long Liang Zhang-Bee-Chung Chen Deepak AgarwalWeiwei Guo, Huiji Gao. Deep natural language processing for search and recom-mender systems. 2019

[4] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu,An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems,IEEE Transactions on Industrial Informatics (Volume: 10 , Issue: 2 , May 2014)

[5] Badsha, S., Yi, X. Khalil, I. Data Sci. Eng. (2016) 1: 161. https://doi.org/10.1007/s41019-016-0020-2

[6] Hung-Wei Chen, Yi-Leh Wu, Maw-Kae Hor, Cheng-Yuan Tang, Fully Content-Based Movie Reccomender System with Feature Extraction Using Nueral Network. Proceedings of the 2017 International Conference on Machine Learning and Cybernetics, Ningbo, China, 9 - 12 July 2017. 978-1-5386-0408-3/17

[7] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. TMIS 6, 4 (2016), 13.

[8] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The Youtube video recommendation sys- tem. In Proceedings of the Recsys. 293–296. http://doi.acm.org/10.1145/1864708.1864770

[9] Rishabh Ahuja,Arun Solanki,Anand Nayyar. Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor. January 2019. DOI: 10.1109/CONFLUENCE.2019.8776969

[10] Baolin Yi, Xiaoxuan Shen*, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. Expanded autoencoder recommendation framework and its application in movie recommendation. 2016 10th International Conference on Software, Knowledge, Information Management Applications (SKIMA). 978-1-5090-3298-3/16

[11] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

[12] Zihuan Wang, Kyusup Hahn, Youngsam Kim, Sanghyup Song, Jong-Mo Seo. A news-topic recommender system based on keywords extraction. Multimed Tools Appl (2018) 77:4339–4353, https://doi.org/10.1007/s11042-017-5513-0

[13] Mir Saman Tajbakhsh, Jamshid Bagherzadeh. Microblogging Hash Tag Recommendation System Based on Semantic TF-IDF. 2016 4th International Conference on Future Internet of Things and Cloud Workshops. DOI=http://dx.doi.org/10.1109/W-FiCloud.2016.59, 978-1-5090-3946-3/16

[14] Kunal Shah, Akshaykumar Salunke, Saurabh Dongare, Kisandas Antala. Recommender Systems: An overview of different approaches to recommendations. 2017 International Conference on Innovations in information Embedded and Communication Systems (ICIIECS). 978-1-5090-3294-5/17

[15] Sanket Doshi. "Brief on Recommender Systems".Feb 10 2019. https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd

[16] Bai, Xiaomei Wang, Mengyang Lee, Ivan Yang, Zhuo Kong, Xiangjie Xia, Feng. (2019). Scientific Paper Recommendation: A Survey. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2890388.

[17] J. Konstan J. Ben Schafer and J. Riedl. Recommendersystems in ecommerce' in proc. pages 158–166. 1st ACM Conf. Electron. Commerce, Denver, CO,USA, 1999.

[18] A S Girsang, B Al Faruq, H R Herlianto, and S Simbolon. "collaborative recommendation system in users of anime films"."J.Phys Conf Ser",("1566 012057"), 2020.

[19] Diao, Q., Qiu, M. Wu, C. Y., MBla, A.J., Jiang, J., Wang, C. (2014 August). Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 193 - 202), 2014.

[20] Uluyagmar, M., Cataltepe, Z., Tayfur, E. (2012). Content-based movie recommendation using different feature sets. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1, pp 17-24), 2012.

[21] Lehinevy, T., Kekkinis-Ntrenis N., Siantikos, G., Dogruoz, A. S., Giannakopoulos, T., Konstantopoulos, S. (2014, November). Discovering similarities for content-based recommendation and browsing in multimedia collections. In Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on (pp. 237-243), 2014.

[22] Donghui Wang, Yanchun Liang, Dong Xu, Xiaoyue Feng, Renchu Guan, A content-based recommender system for computer science publications, Knowledge-Based Systems, Volume 157, 2018, Pages 1-9, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2018.05.001. (https://www.sciencedirect.com/science/article/pii/S0950705118302107)

[23] B. Chen, H. He, J. Guo, Constructing maximum entropy language models for movie review subjectivity analysis, J. Comput. Sci. Technol. 23 (2) (2008) 231–239.

[24] N. Gupta, P. Saxena, J. Gupta, Document summarisation based on sentence ranking using vector space model, Int. J. Data Min. Model. Manag. 5 (4) (2013) 380–406.

[25] R. Costa, C. Lima, Document clustering using an ontology-based vector space model, Int. J. Inf. Retr. Res. 5 (3) (2015) 39–60.

[26] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito. Exploiting the Web of Data in Model-based Recommender Systems. RecSys'12, September 9–13, 2012, Dublin, Ireland. (2012) ACM 978-1-4503-1270-7/12/09

[27] Thushara, M. G., Tadi Mownika, and Ritika Mangamuru. "A comparative study on different keyword extraction algorithms." 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2019.

[28] Thushara MG, Krishnapriya MS, and Sangeetha S Nair. Domain classification and tagging of research papers using hybrid keyphrase extraction method. 06 2017.

[29] Thushara, M. G., M. S. Krishnapriya, and Sangeetha S. Nair. "A model for auto-tagging of research papers based on keyphrase extraction methods." 2017 International conference on advances in computing, communications and informatics (ICACCI). IEEE, 2017.

[30] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. Text Mining: Applications and Theory, pages 1–20, 2010.

[31] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

[32] Prajit Datta. MovieLens 100K Dataset. Stable benchmark dataset. 100,000 ratings from 1000 users on 1700 movies. https://www.kaggle.com/prajitdatta/movielens-100k-dataset.