ISSN 2063-5346

A NOVEL RECOMMENDER SYSTEM EXPLOITING COLLABORATIVE FILTERING MODEL BASED ON KEYWORD EXTRACTION FROM TEXT

VIJAYA SHETTY S[1], KHUSH DASSANI[1], HARISH GOWDA G.P[1], HARIPRASAD REDDY P[1], SEHAJ JOT SINGH[1], SAROJADEVI H[1]

Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India[1]

Article History: Received: 03.04.2023 **Revised:** 25.04.2023 **Accepted:** 15.05.2023

Abstract

The tremendous growth of the Internet and the rise of social media has helped many organizations to fortunately collect voluminous data about its' customer base. The data pertaining to customer base are the most influenceable and valuable resources for the organizations at the present time. Organizations may utilize these data to produce high revenues out of the business. Various mining techniques can be used on these data to gather valuable insights which can be used by the organization to make fascinating recommendations to their customers. These recommendations can bring in increased sales and profits to the organization. Content based recommendation is one such techniques that use item features to calculate item similarities and make recommendations. However, such techniques in a production environment use only categorical data instead of the full-fledged item-descriptions or item-review data because of their large size and increased computational resources. In this paper we provide a novel keyword-based approach for book recommendation. The text conversion technique used in the research tries to effectively reduce the text corpora while keeping the valuable information and, provides a content-based user-item rating and, fuses it with a Singular Value Decomposition (SVD) trained model to generate content lenient collaborative filtering user-item rating. The results obtained reveal that the collaborative filtering approach with keyword description has the lowest Root Mean Square Error(RMSE) score of 0.59 which is significantly lower than the RMSE score of 0.86 of collaborative filtering.

Keywords : Recommender system, Collaborative filtering model, Keyword extraction, Natural language processing, Text embeddings.

1.Introduction

Recommender systems are instrumental tools for organizations to improve financial gain and develop a competitive edge on their competitors using the data collected from the customers. Large commercial enterprises like Amazon, YouTube and Netflix have invested considerably to build cutting edge recommendation systems that assisted them to be promoted as giant companies. Recommender systems not only cater the customer base more efficiently but also improve company's business value in the long run . Recommender systems provide an additional layer of service to existing business. Building a recommender system is very challenging because they must handle various factors like frequently changing customer behavior, sparse data, time relevance, demographic trend and cold start problem. Besides, it is hard to find the accuracy of recommender systems as it is difficult to judge the usefulness of the recommendations made by the system.

Today recommender systems have evolved as a widely used machine learning process used to increase product sales, for better product generation, provide better customer experience, better ads recommendations and many more. There are several content-based, collaborative-filtering based and hybrid techniques in which a recommender system can be designed and based on one's business needs, careful selection among these techniques should be made.

Collaborative filtering systems make use of the ratings given by the customers to give recommendations. The ratings may be explicit rating or implicit rating. Explicit ratings are normally of varying scales. The difficulty associated with explicit rating is that the rating is sparse because users may not give ratings for all of the items purchased. Implicit ratings are normally based on factors such as the clicks, visits etc. that are associated with user interactions. In implicit ratings the item is rated as 1 in case the user has viewed the item, a 0 otherwise. Such information can be drawn from the web logs. There exist two methods for developing a collaborative filtering recommender system. First one is a memory-based approach and the second is a model-based approach. Memory-based approaches locate users with similar preferences as of the target user and are relatively simple to implement. In model-based approach, a model is built for the rating matrix to provide recommendation to customers. Model-based approaches are of 2 types; the first type is user-based and the second is item-based. The user-based recommender system represents each user in the dataset as a vector of item ratings and the distance between user vectors are measured using appropriate distance measure, Euclidian distance for instance. The users with trivial vector distances are considered to have similar preferences. The item-based recommender system represents each item in the dataset as vector of user ratings and similarities between the item vectors are found to identify items with similar ratings. Recommendations is based on the item-item similarity which is calculated as the weighted sum of the user ratings. Scalability of item-based models are better than the user-based models and are normally preferred over the user-based models. In essence, the collaborative filtering method is significantly affected by the changing user preferences, inclusion of new users/ new items and the popular item list.

In content-based models, the categorical features of the items are used to make product recommendations. If the full-fledge item description or item reviews are used they result in storage issues as well as increased computational time, which calls for a technique that can effectively store most of the information in the text data while reducing the text size. On the other hand, in collaborative filtering[1][2][3] approaches, Matrix Factorization (MF) is an approach which is implemented in a variety of ways to achieve effective business goals. The SVD is a model-based MF technique which decomposes a user-item rating matrix into latent features of users and items to calculate the sparse rating matrix. However, these SVD is mathematical technique which is based on rank reduction that does not take any item content features into account while making recommendations. So, fusion of both content based and SVD recommenders is necessary to generate a rating that not only considers the user-interests but also the item content.

The objective of the proposed research is to include the product information based on keywords along with the user ratings to make personalized recommendations in a computationally effective way.

2.Related work

Hybrid recommender systems fuse multiple recommender systems exploiting the benefit of each system in order to deliver superior recommender systems. In general the collaborative filtering and the content-based modules are bundled together to avoid their limitations. Despite the fact that a hybrid recommender system leverages the benefits of divergent algorithms, devising a Hybrid recommender system is a very demanding task. There exist distinct approaches to build a Hybrid recommender system.

Hybrid recommender systems using Matrix Factorization can be implemented in a variety of ways. The MF techniques when provided with side information like item features provides improved results. These information can be incorporated in two methods. In the first method used by W. X. Zhao and P. S. Yu, 2019[4] and “Capsmf”, 2020[5][6], item similarities are calculated separately and then extended for use into the MF models. In the second method used by Zhang et al, 2021[7][8] features are directly included into the item latent feature matrix obtained from MF models. These technique scales linearly with the dataset and thus reduces computation complexity. This technique considers features of items as item-feature relations and tries to project it in the same space as that of the user-latent feature and the item-latent feature space. Information can also be included to better understand the structural and semantic information about the textual data into the MF techniques (W. X. Zhao and P. S. Yu, 2019)[4]. MF techniques can also be further extended to use encryption techniques that preserves the privacy of the user (Ogunseyi et al, 2021)[9].

Deep learning is another field where textual data is used for rating prediction. RBM or Restricted Boltzmann Machines can be boosted with side information such as user’s demographic to better predict the users’ ratings (Chen Z et al, 2020)[10][11]. Convolution neural network are the most widely used technique while dealing with textual data because of their ability to extract features from data. CNN can be used to extract sematic relationship in text which can be beneficial in rating prediction (R Cao et al, 2020)[12]. CNN can be trained to generate user preferences and item properties from text to generate rating prediction (Lei Zheng et al, 2017)[13][14]. CNN can be also used to generate concise review from actual text which can greatly reduce computing overhead while making predictions (Y. -C. Chou, 2020)[15]. Deep learning techniques can also be fused with MF techniques for rating prediction which can be beneficial. MF can be used to learn the linear information from the user-item interaction and the deep learning techniques can be used to include to nonlinear information form user-item reviews (C. Wei et al, 2021)[16].

Keyword extraction is an automated technique that extracts key information from a large corpus of text focusing on which words are more relevant than others. Again, a lot of research is done in these field. TextRank[17] is the most widely used technique for extraction of keywords from text and is based on the PageRank algorithm. The results of TextRank may vary depending on the co-occurrence window size, iteration number and decay factor (M. Zhang et al, 2020)[18][19]. YAKE[20] is another tool which unlike most other keyword extraction tools, which depends on a large, annotated text corpus, uses unsupervised learning based on statistical approaches to extract keywords from a single document.

3.Experimentation

A hybrid recommender system has been implemented, in which a rating of an item for a particular user is found using both content information and user-item interaction. Both the item review and the item description information have been incorporated in the process to include the effect of the nearest neighbors of the item already rated by the user. The Goodreads dataset generated through scrapy is employed in the research.

3.1. Modules Used

The following subsections describe the techniques used in the experimentation.

3.1.1. Natural Language processing

Natural language processing(NLP) is actually a machine learning process concerned with using text in a way which makes it easier for machines to understand. NLP is a combination of rule-based modelling with statistical, machine learning and deep learning models. NLP helps us to process text in a much easier way by providing various libraries for text cleaning. In text the common words like 'the', 'is', 'are' etc. are just to provide structure. They do not have much information about the actual content and hence must be removed to perform any machine learning task. NLP also helps in the process of lemmatization which is a text normalization technique. Lemmatization is essential for converting any word to its root form. Lemmatization helps in better predicting text similarity than text which are not lemmatized.

3.1.2. Keyword extraction

Keyword extraction is an essential part of our experiment as it helps us in reducing the text data by retrieving the most useful information from text data. It not only reduces the text corpus but also keeps the information in item description and item reviews intact. Yake is used for keyword extraction in the experiment which is lightweight unsupervised tool for keyword extraction which used statistical approach for finding keywords from a single document. This is much more suitable for this experiment as it does not depend on a pool of large, annotated text which must be available all the time.

3.1.3. Text Embeddings

Text embedding is another useful technique implemented in this research for fast calculation of item similarities [21]. In text embeddings, text is converted into vectors called word embeddings. Text embedding is a way of converting text into numerical representation which captures information like analogies or semantic meaning. In the experiment a token-based text embeddings trained on English Google News 200B corpus[22] from TensorFlow is used. These embeddings make calculation of item similarity faster without having to worry about loss of valuable information.

3.1.4. Cosine similarity

Cosine similarity helps us to find how close the text embeddings are in space. More the cosine similarity, more identical the text [23][24]. These similarity mechanism helps us predict which item is more like the target item, thus helping in content-based recommendation. Formulae for calculating cosine similarity of two text embeddings A and B is given in eqn. (1)

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

where A.B = dot product of the vector 'A' and the vector 'B', ||A|| and ||B|| = length of the vectors 'A' and 'B' and ||A|| ||B|| = cross product of the vector 'A' and the vector 'B'.

3.1.5. SVD

Singular value decomposition or SVD [25] is a model-based MF technique which takes the user-item rating matrix and decomposes the matrix into two latent feature matrices based on a mathematical technique called rank reduction. The two matrices generated are the so-called user-latent feature matrix and the item-feature matrix which nothing but description of each user and each item in the rating dataset in terms of hidden features obtained from the dataset generated by a technique called Principal Component Analysis or PCA. Thus, SVD tries to reduce the unnecessary parameters in the dataset by finding correlations in the rating matrix. These latent feature matrices can help us by filling the original sparse rating matrix for rating prediction. In the experiment TensorFlow is used for implementation of the SVD model which gives us both the user-latent embeddings and item-latent embeddings for rating prediction.

3.1.6. TensorFlow

TensorFlow is an open-source tool for machine learning which provides a variety of flexible and easy to use libraries for research purposes. It provides workflows both in Python and JavaScript and makes writing code for machine learning purposes a lot easier. It implements a lot of the algorithms out of the box so that one can focus on using these algorithms in their own way without writing the code from scratch for each of the experiment.

3.2. Workflow Design

Figure 1. explains the workflow view of the experiment. From the description dataset only those books having an English description or English title are extracted. The books which may be considered as unreasonable to recommend are also removed based on the avoiding stop list. Then the natural language processing is applied to the text to perform text cleaning, removing stop words and performing lemmatization. Then the keywords from each description are extracted using Yake model. The book descriptions based on the keywords extracted are regenerated. After these steps there will be two representation of the item description – one is only cleaned, and lemmatized description called processed description and the other is keyword-based description called the keyword description. From the reviews dataset we segregate books with English description and extract keywords for each review of the book. All the review keywords for a particular book and reduce the set of keywords based on the average score are then joined. Thus, a reduced set of keywords from reviews is obtained. These reduced review keywords are combined with the keyword-based description to obtain a mix representation of an item.

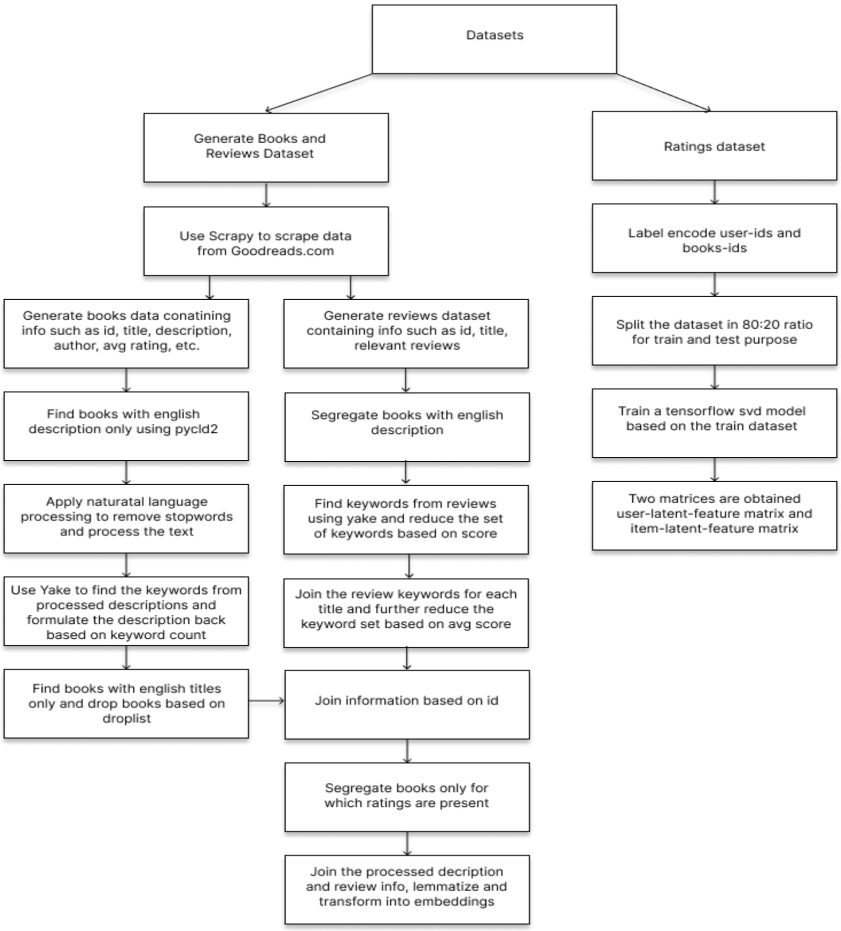

Datasets
Generate Books and Reviews Dataset
Ratings dataset
Use Scrapy to scrape data from Goodreads.com
Label encode user-ids and books-ids
Generate books data conatining info such as id, title, description, author, avg rating, etc.
Generate reviews dataset containing info such as id, title, relevant reviews
Split the dataset in 80:20 ratio for train and test purpose
Find books with english description only using pycld2
Segregate books with english description
Train a tensorflow svd model based on the train dataset
Apply naturatal language processing to remove stopwords and process the text
Find keywords from reviews using yake and reduce the set of keywords based on score
Two matrices are obtained user-latent-feature matrix and item-latent-feature matrix
Use Yake to find the keywords from processed descriptions and formulate the description back based on keyword count
Join the review keywords for each title and further reduce the keyword set based on avg score
Find books with english titles only and drop books based on droplist
Join information based on id
Segregate books only for which ratings are present
Join the processed decription and review info, lemmatize and transform into embeddings

Figure 1 Workflow Diagram

As shown in table 1, there are four content representation for an item, one book for instance - processed description, keyword description, review representation and mix representation.

Table 1. Four content representation for a particular book in the dataset

Type	**Example**
Processed Description	wharton final novel completed marion mainwaring author death revolves around american british society s told large part eye american debutante story portrays innocent wide eyed almost ethereal girl turn socially conscious woman financial worry unrecognizable even beginning section quickly catch listener attention lush description room clothes height feminine beauty enter world intrigue secret character past relationship could prove fatal competition taken limit literary value notwithstanding book might appeal soap opera romance fan attentive listener quickly becomes disconcerting character awkward british sounding name added increasingly difficult recall without backing tape library pas one rochelle ratner formerly poetry editor soho weekly news new york
Keyword Description	marion mainwaring american british told american quickly characters quickly characters british rochelle ratner york

Review Representation	wharton buccaneers end testvalley guy miss completely
Mix Representation	marion mainwaring american british told american quickly character quickly character british rochelle ratner york wharton buccaneer end testvalley guy miss completely

From the above dataset. only the books which are present in both the item representation dataset and the rating dataset are extracted. From the ratings dataset, an 8:2 train-test split is created. The SVD model is then trained on the train dataset.

3.3. Pseudo Code for rating prediction

Figure 2. explains the process of generating the rating for a book with respect to a particular user. Table 2 gives the pseudocode for the generation of ratings. For rating prediction of an item for a user the first step is to find the top-1000 items similar to the user based on the item description. From the top-1000 similar items only those items whose similarity score is greater than 0.5 are considered. Next find the items which are already rated by the user and remove the items which have a similarity score less than the average similarity score. Using those items try to formulate a content_score calculated as the dot product of item similarity and item rating. Next find the svd_predicted_rating of the item using the trained SVD model. Now the model_rating is calculated using eqn. (2).

$$model\,rating = 0.5 \times \text{content\_rating} + \ 0.5 \times \text{svd\_predicted\_rating} \qquad (2)$$

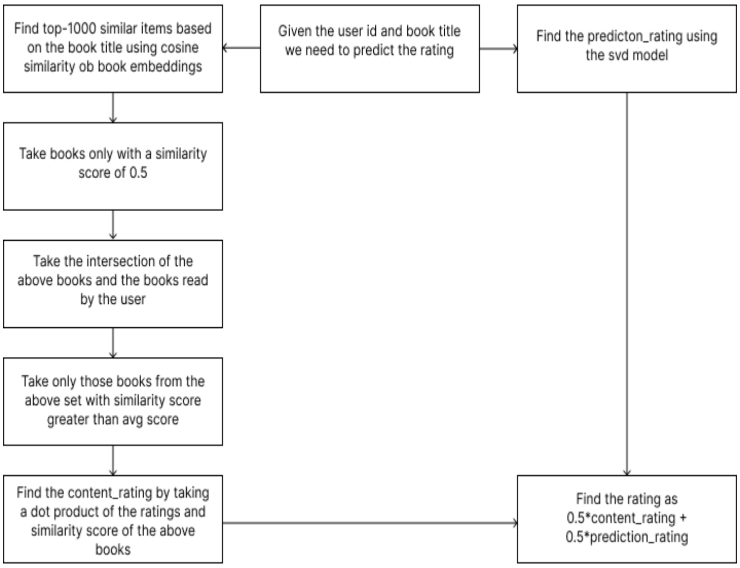

Find top-1000 similar items based on the book title using cosine similarity ob book embeddings
Given the user id and book title we need to predict the rating
Find the predicton_rating using the svd model
Take books only with a similarity score of 0.5
Take the intersection of the above books and the books read by the user
Take only those books from the above set with similarity score greater than avg score
Find the content_rating by taking a dot product of the ratings and similarity score of the above books
Find the rating as 0.5*content_rating + 0.5*prediction_rating

Figure 2 Process of generating the rating for a book w.r.to one User

Table 1. Pseudocode for rating prediction

Given a user U and item I predict rating(U, I) -

1. for the item I find top-1000 similar items {i1, i2, i3, i4, ………, i1000}
2. Reduce the set by elimination of the items with similarity score < 0.5 so the set becomes Reduced{i1, i2, i3, i4, ………, i1000}
3. Find the items rated by the User U - {u1, u2, u3, u4, ………, uk} where k denotes the count of items rated by the customer/user U.
4. Find the intersection of the sets Reduced{i1, i2, i3, i4, ………, i1000} and {u1, u2, u3, u4, ………, uk} - Intersect{x1, x2, x3, ………, xn}.
5. Eliminate those items from the intersect set with a similarity score less than the average similarity score so the set becomes - Reduced-Intersect {x1, x2, x3, ………, xn}.
6. Now, using dot product of the similarity score and the user ratings for the item in the Reduced- Intersect{x1, x2, x3, ………, xn} set calculate the content_rating score.
7. Calculate the svd_predicted_rating using the trained SVD model.
8. Output the model rating as –

   Model_rating = 0.5×content_rating + 0.5×svd_predicted_rating.

4.Results And Discussions

To compare the performance and accuracy of different models two of the most commonly used metrics are MAE and RMSE.

In order to draw a comparative analysis of our models we calculate the RMSE score of each of the following models on our test dataset - Content based model using keyword description, Content based model using review representation, Content based model using processed description only, Content based model using mix representation, Collaborative filtering, Collaborative filtering with keyword description, Collaborative filtering with review representation, Collaborative filtering with processed description and Collaborative filtering with mix-representation. Equation 3 shows the formula for RMSCE calculation. RMSE computes the standard deviation of the prediction errors.

$$RMSE = \sqrt{\frac{1}{|R|}\sum_{p(r)\in R} \left(r_{ui} - p(r)_{ui}\right)^2} \quad (3)$$

Here, r_{ui} and $p(r)_{ui}$ are the actual and the predicted ratings and, R is the test data. RMSE computes how distant are these errors from the line of best fit. RMSE highly penalizes bad predictions due to the square, hence it is highly affected by bad predictions. Lower the RMSE better the prediction of the model. Table 3 shows the comparison of different models by RMSCE. Figure 3 shows the graph of the same.

Table 3. Model comparison

Model	RMSE SCORE
Content based model using keyword description (CBD)	0.62
Content based model using review representation (CBR)	0.97
Content based model using mix representation (CBDR)	1.04
Collaborative filtering (CF)	0.86
Collaborative filtering with keyword description (CFD)	0.59
Collaborative filtering with review representation (CFR)	0.78
Collaborative filtering with mix-representation (CFDR)	0.79
Content based model using processed description (CBN)	1.23
Collaborative filtering with processed description (CFN)	0.94

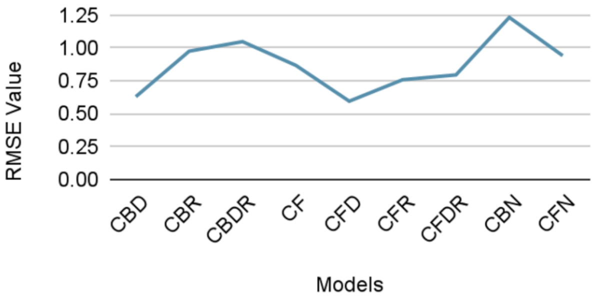
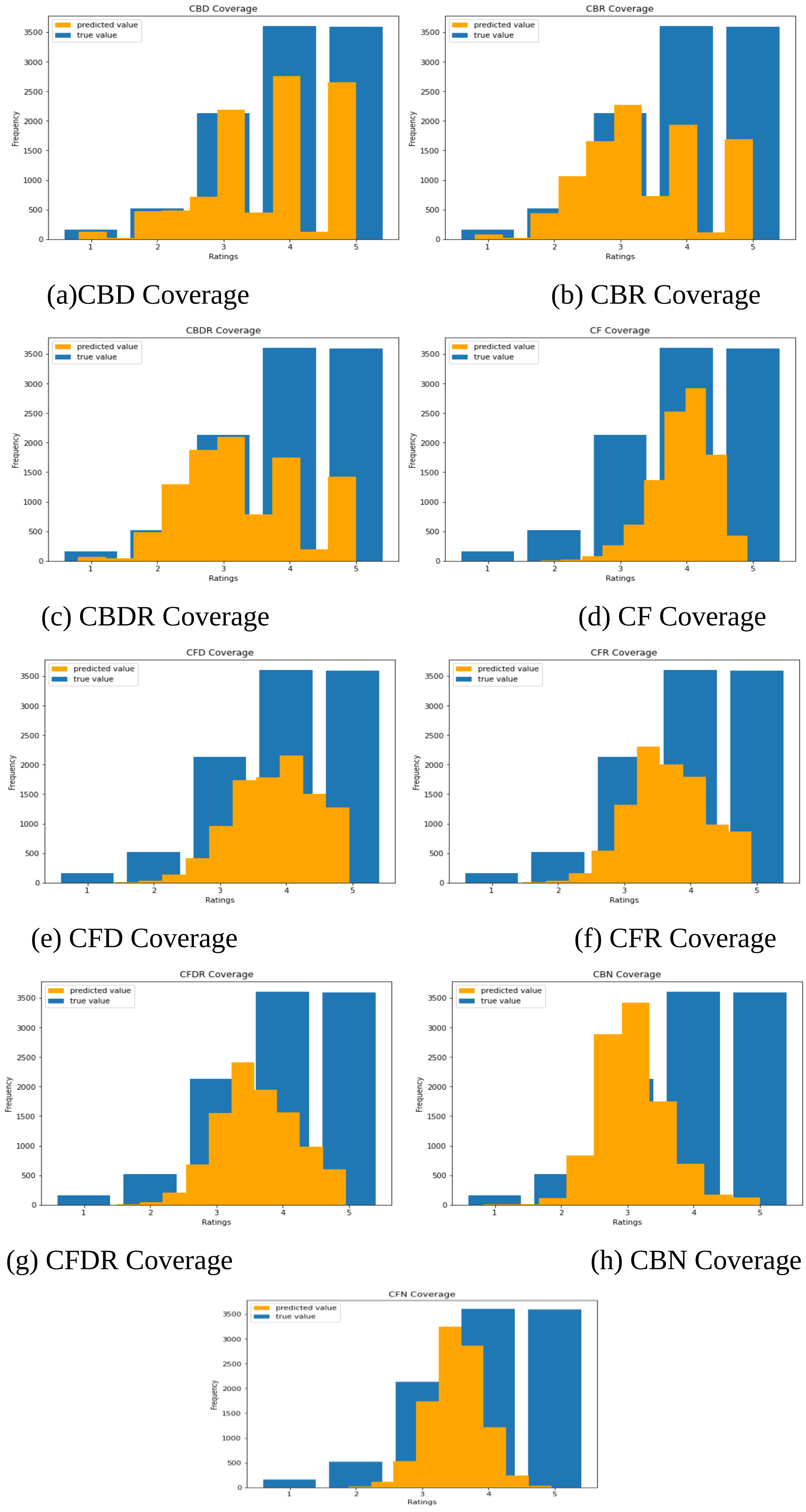


Figure 3 Comparison of various models

Figure 3 shows that the use of keyword description with user-item rating matrix greatly improves the performance of the collaborative filtering model. The collaborative filtering approach with keyword description has the lowest RMSE score of 0.59 which is significantly lower than the RMSE score of 0.86 of collaborative filtering. Moreover, using token based reformed description improves the performance of the collaborative model as opposed to processed descriptions which degrades the performance.

The performance of each model is depicted in figure 4 (a)-(i).



(a)CBD Coverage (b) CBR Coverage

(c) CBDR Coverage (d) CF Coverage

(e) CFD Coverage (f) CFR Coverage

(g) CFDR Coverage (h) CBN Coverage

(i) CFN Coverage

Figure 4. Model Comparison

Figure 4(a)-(i) illustrates the ratings of books in the test data and the predicted ratings of each model. The X-axis represents ratings and the Y-axis shows the number of books. The blue graph represents the actual ratings of the test dataset, which are more interval values [1, 2, 3, 4, 5] and the graphs in yellow represent predicted ratings which are more continuous value 1-5. There are 9 different performance graphs, one for each model. While reviewing this graph, one has to observe the density around each of the ratings 1, 2, 3, 4 and 5. In the above graphs it is clearly seen that the CFD graph best replicated the original test data w.r.to. density distribution of ratings and it is concluded that the SVD model gives the best result when it is mixed with keyword-based description for rating predictions.

Data storage has been a huge problem for any IT company as it requires investing in huge datacentres or using expensive cloud-based services. Storing large corpus of product related textual data and tons of customer reviews are generated is essential as they can drive a business to success. Using the keyword-based technique proposed by our studies one can greatly reduce the text corpora helping the companies in data storage in an efficient way. Moreover, processing of huge text data requires expensive machines and architectures putting a lot of loads on resources. The proposed keyword-based technique ensures a computationally inexpensive process with less loads on resources.

4.1. Limitations

In the proposed hybrid recommender system, a 50:50 split of both content based and collaborative based filtering ratings has been considered for recommendation. However, this ratio does not consider which factor is more accountable for user taking an action. Therefore, this rating can be improved by considering this split based on users' decision-making processes. Although we have fused multiple text information together directly, there can be a better approach of doing it like taking into consideration of which text actually causes the user to take better decisions.

5.Conclusions

This research is an attempt to improve the performance of collaborative filtering by the inclusion of item information such as description and item reviews. Performance improvement not only means improving the RMSE but also the computation time. Using keyword based reformed description improves the performance of the collaborative model as opposed to processed descriptions which degrades the performance. Moreover, using Yake to extract keywords and reconstruction of item description helps us to eliminate large amounts of textual data. Reducing the keyword set from the reviews based on the score given by the Yake tool also helps to eliminate the large corpus of reviews that may be available for an item. In the above report using item description with user-item rating matrix greatly improves the performance of the collaborative filtering table. The collaborative filtering approach with keyword description has the lowest RMSE score of 0.59 which is significantly lower than the RMSE score of 0.86 of collaborative filtering. In fact, in each of the three cases where item description is used with collaborative filtering the RMSE score improves. In future we can also try to leverage the item similarity based on the item latent features obtained using the SVD model.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. “Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges”. ACM Computing Surveys, vol 47, Issue 1, July 2014, Article No.: 3, pp 1–45. https://doi.org/10.1145/2556270

[2] Srifi, M.; Oussous, A.; Ait Lahcen, A.; Mouline, S. “Recommender Systems Based on Collaborative Filtering Using Review Texts—A Survey”. Information 2020, 11, 317. https://doi.org/10.3390/info11060317

[3] Roy, D., Dutta, M. “A systematic review and research perspective on recommender systems”. J Big Data 9, 59 (2022). https://doi.org/10.1186/s40537-022-00592-5

[4] C. Shi , B. Hu, W. X. Zhao and P. S. Yu, "Heterogeneous Information Network Embedding for Recommendation," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 2, pp. 357-370, 1 Feb. 2019, doi: 10.1109/TKDE.2018.2833443.

[5] Katarya, R., Arora, Y. “Capsmf: a novel product recommender system using deep learning based text analysis model”. Multimed Tools Appl, vol 79, Issue 47-48, Dec 2020 pp 35927–35948, https://doi.org/10.1007/s11042-020-09199-5.

[6] Yunfei He, Yiwen Zhang, Lianyong Qi, Dengcheng Yan, Qiang He,”Outer product enhanced heterogeneous information network embedding for recommendation”, Expert Systems with Applications,Volume 169, 2021, 114359, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.114359.(https://www.sciencedirect.com/science/article/pii/S095741742031040X).

[7] H. Zhang, I. Ganchev, N. S. Nikolov, Z. Ji and M. O’Droma, " FeatureMF: An Item Feature Enriched Matrix Factorization Model for Item Recommendation," in IEEE Access, vol. 9, pp. 65266-65276, 2021, doi: 10.1109/ACCESS.2021.3074365.

[8] Idris Rabiu, Naomie Salim, Aminu Da'u, Akram Osman, Maged Nasser,”Exploiting dynamic changes from latent features to improve recommendation using temporal matrix factorization”, Egyptian Informatics Journal,Volume 22, Issue 3,2021,Pages 285-294,ISSN 1110-8665,https://doi.org/10.1016/j.eij.2020.10.003.(https://www.sciencedirect.com/science/article/pii/S1110866520301584)

[9] T. B. Ogunseyi, C. B. Avoussoukpo and Y. Jiang, "Privacy-Preserving Matrix Factorization for Cross-Domain Recommendation," in IEEE Access, vol. 9, pp. 91027-91037, 2021, doi: 10.1109/ACCESS.2021.3091426.

[10] Chen, Z., Ma, W., Dai, W., Pan, W., & Ming, Z. (2020). “Conditional restricted boltzmann machine for item recommendation”. Neurocomputing, vol 385, 269-277. doi:10.1016/j.neucom.2019.12.088.

[11] Reza Shafiloo, Marjan Kaedi, Ali Pourmiri. “Predicting user demographics based on interest analysis”, Human-Computer Interaction.https://doi.org/10.48550/arXiv.2108.01014.

[12] R. Cao, X. Zhang and H. Wang, "A Review Semantics Based Model for Rating Prediction," in IEEE Access, vol. 8, pp. 4714-4723, 2020, doi: 10.1109/ACCESS.2019.2962075.

[13] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. “Joint Deep Modeling of Users and Items Using Reviews for Recommendation”. In Proceedings of the Tenth ACM International

Conference on Web Search and Data Mining (WSDM '17). Association for Computing Machinery, New York, NY, USA, 425–434. doi:https://doi.org/10.1145/3018661.3018665.

[14] Li Q, Li X, Lee B, Kim J. A Hybrid CNN-Based Review Helpfulness Filtering Model for Improving E-Commerce Recommendation Service. Applied Sciences. 2021; 11(18):8613. https://doi.org/10.3390/app11188613

[15] Yun-Cheng Chou, Hsing-Yu Chen, Duen-Ren Liu And Der-Shiuan Chang, "Rating Prediction Based on Merge-CNN and Concise Attention Review Mining," in IEEE Access, vol. 8, pp. 190934-190945, 2020, doi: 10.1109/ACCESS.2020.3031621.

[16] C. Wei, J. Qin and W. Zeng, "DNR: A Unified Framework of List Ranking With Neural Networks for Recommendation," in IEEE Access, vol. 9, pp. 158313-158321, 2021, doi: 10.1109/ACCESS.2021.3130369.

[17] Ning Zhou, Wenqian Shi, Renyu Liang, Na Zhong, "TextRank Keyword Extraction Algorithm Using Word Vector Clustering Based on Rough Data-Deduction", Computational Intelligence and Neuroscience, vol. 2022, Article ID 5649994, 19 pages, 2022. https://doi.org/10.1155/2022/5649994

[18] M. Zhang, X. Li, S. Yue and L. Yang, "An Empirical Study of TextRank for Keyword Extraction," in IEEE Access, vol. 8, pp. 178849-178858, 2020, doi: 10.1109/ACCESS.2020.3027567.

[19] Weifeng Zhang, "Management and Plan of Undergraduates' Mental Health Based on Keyword Extraction", Journal of Healthcare Engineering, vol. 2021, Article ID 3361755, 9 pages, 2021. https://doi.org/10.1155/2021/3361755

[20] Campos, Ricardo & Mangaravite, Vítor & Pasquali, Arian & Jorge, Alípio & Nunes, Célia & Jatowt, Adam. (2020). "YAKE! Keyword Extraction from Single Documents using Multiple Local Features". Information Sciences. vol 509. 257-289. doi:10.1016/j.ins.2019.09.013.

[21] Mishra, Asha & Panchal, V & Kumar, Pawan. (2020). "Similarity Search based on Text Embedding Model for detection of Near Duplicates". International Journal of Grid and Distributed Computing. 13. 1871-1881.

[22] Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, Yu Sun,"Pre-Trained Language Models and Their Applications",Engineering,2022,ISSN 2095-8099,https://doi.org/10.1016/j.eng.2022.04.024. (https://www.sciencedirect.com/science/article/pii/S2095809922006324)

[23] M. Lerato, O. A. Esan, A. Ebunoluwa, S. M. Ngwira and T. Zuva, "A survey of recommender system feedback techniques, comparison and evaluation metrics," International Conference on Computing, Communication and Security (ICCCS), 2015, pp. 1-4, doi: 10.1109/CCCS.2015.7374146.

[24] Vijaya Shetty, S., Dassani, K., Harish Gowda, G.P., Sarojadevi, H., Hariprasad Reddy, P., Singh, S.J. (2022). "A Review of the Techniques and Evaluation Parameters for Recommendation Systems". In: Raj, J.S., Shi, Y., Pelusi, D., Balas, V.E. (eds) Intelligent Sustainable Systems. Lecture Notes in Networks and Systems, vol 458. Springer, Singapore. https://doi.org/10.1007/978-981-19-2894-9_35

[25] Bipul Kumar,"JIA NOVEL LATENT FACTOR MODEL FOR RECOMMENDER SYSTEM", Journal of Information Systems and Technology Management,Vol. 13, No. 3, Set/Dez., 2016 pp. 497- 514,ISSN online: 1807-1775,DOI: 10.4301/S1807-17752016000300008

Biographies of Authors

Vijaya Shetty S is currently working as Professor and Head in the Department of Computer Science and Engineering, at Nitte Meenakshi Institute of Technology, Bangalore. Her research interests include Data Mining, Machine Learning, Deep Learning and Distributed Computing. She is a Life member of Indian Society for Technical Education (ISTE) and member of Computer Society of India (CSI). She can be contacted at email: vijayashetty.s@nmit.ac.in

Khush Dassani is currently working as Software Engineer at Cimpress. As a Software Engineer, he develops tools to facilitate easy and effective brand building. His qualifications include holding a B.E. degree in Computer Sciences and Engineering from Nitte Meenakshi Institute of Technology, Bangalore.

Harish Gowda GP is currently working as a Software Engineer at ShipX. In his current role, he works with a team of engineers to develop software solutions in supply chain management that help businesses operate more efficiently and effectively. He obtained his Engineering degree from Nitte Meenakshi Institute of Technology, Bangalore.

Hariprasad Reddy P is currently working as a Data Scientist at Innover Digital, an American startup. In his current role as a Data Scientist, he collaborates with his team to provide software solutions with the help of data analytics and machine learning, as well as develop and deploy appropriate models to allow businesses to capitalize on current trends and technologies, allowing them to become future-ready organizations. He obtained his Bachelor's degree from Nitte Meenakshi Institute of Technology, Bangalore.

Sehaj Jot Singh is currently working as Software Engineer at Accenture. As a software engineer, he analyzes and modifies existing software, and design, construct and test end-user applications that meet user needs — all through software programming languages. He obtained his Bachelor's degree from Nitte Meenakshi Institute of Technology, Bangalore.

Dr. Sarojadevi H. is a professor in the department of CSE at NMIT, Bangalore, India. She is a PhD graduate from the Indian Institute of Science, Bangalore. Her areas of interest include Machine learning, Cloud computing, System performance and Natural Language Processing. She has won several national and international recognitions, including Marquis Who's Who in the World 2016. She can be contacted at email: sarojadevi.n@nmit.ac.in