Content Based Filtering Algorithm for Computer Science Research Article Recommendation

Sivasankari.R[1] and Dhilipan.J[2]

[1] Research Scholar, Department of Computer Applications (MCA), SRM Institute of Science and Technology, Ramapuram, Chennai, 600 089, India.
[2] Professor & Head, Department of Computer Applications (MCA), SRM Institute of Science and Technology, Ramapuram, Chennai, 600 089, India.
manjurmoni@gmail.com

Abstract. It is common to find recommendation systems these days since the Internet has so much digital information uploaded every day. This overload of digital information makes it hard to find the specific information needed quickly. Recommendation systems are used to offer users a list of suggestions that may be required for user based on the data they provide. The fundamental purpose of these recommendation systems is to predict what the user might be interested in. Recommendation systems can be used in various fields, such as entertainment, shopping, business, etc. With the increased number of research articles, it is important for the researchers to have a recommendation system that makes searching for relevant articles simpler for the researcher. In this paper, we describe a recommendation system that uses proposed CBRAlgorithm to suggest the top ten papers based on the concept that the user inserts. As a matter of description, this article utilizes the concept of TFIDF to implement vectorization, and it uses a cosine similarity measure to find the relevant papers along with proposed CBRAlgorithm.

Keywords: Research Article Recommendation, TF-IDF, Cosine similarity, Vectorization, Research Article Suggestion, CBRAlgorithm

---

Introduction

A research article recommendation system is used to recommend articles those were relevant to the search query of the user. These recommendation systems usually reduce the time taken by the user to search for his/her topic of interest. The recommendation system can be classified into content-based systems, collaborative systems, graph-based systems, and hybrid systems. The recommendation can be done by considering article features like keyword/title search, Co-author search, and co-citation search. In the case of the keyword search primarily the abstract and keyword of all articles in the database are taken and similarity between those articles with the search query is calculated and the articles with high similarity scores are considered to be relevant for the search query. These articles are then recommended by the model to the user. For measuring the text similarity, the most commonly used algorithms are Jaccard, TF-IDF, Doc2vec, USE, and BERT. The next method to suggest research articles is based on the co-authored recommendations. In this method, the system requires an article that is present in the database as a search query along with the citation details of the paper. From these details, the co-author and co-citation networks are used to find the relevant articles. It finds the articles that have the highest common citations with the queried paper and considers them relevant. Also, in the case of a co-author network, it attempts to find the articles of the author in the query that are relevant and considers them as relevant. In this article, the computer science research articles are recommended for the user by matching the search query by using the CBRAlgorithm and techniques like term frequency-inverse term frequency and cosine similarity.

The contribution of the work in this paper is outlined as follows:

- The text document is preprocessed for getting the accurate result for the recommendation process that includes process like stop words removal, stemming and lemmatization.
- Then proposed CBRAlgorithm is applied to find the vector and compute the similarity of the documents.

- Finally, the relevant documents are recommended by the CBRAlgorithm based on the TFIDF and Cosine Similarity.

The rest of the document is organized as section 2 literature review, section 3 defines the dataset used in the paper, section 4 presents methodologies employed in this along with proposed Content-Based Recommendation Algorithm(CBRAlgorithm) , section 5 describes the experimental setup with results with section 6 concludes the papers.

Literature Review

Yi Li et al.[3] built a PRHN Network to suggest papers to users based on the preferences of the user. They design a heterogeneous network for expressing Object-Relationships in this article, develop a meta-path to get a meta-score, and then recommend Top N articles to the user based on this meta-score[3]. In HRM model, Xinyi Li et al.[5] modeled a feed-forward neural to generate a scoring function to recommend the articles along with content and its behaviors. K., Kan et al.[6] authors construct vectors for 3 different parts first, it computes word list from its paper, second words from papers that cite the original paper, and words list from papers that it refers to. After computing the weight of all these vectors the final value is used for recommending papers [6]. The method proposed by Wang, G et al [7] combined the CBF and CF models along with social information of the article in the SSN platform to compute the weights. Cai X et al.[8] proposed a model to combine a three-layered graph with each layer working with separate objects like papers, researchers, and venues, it uses clustering at each level of the graph. Xia et al [9] propose a method, which includes authors relation between articles to generate recommendations for target users[14] including features of pairwise article ratio and most commonly appeared author's ratio. Son et al [10] devised a three-step methodology that begins with the creation of multilevel citation networks, then moves on to the selection of candidate papers, and finally to the determination of each candidate paper's rating for final recommendation. Zhao et al [12] developed a domain knowledge model–concept map, and background knowledge of the domain to assess the gap in the user knowledge. Background knowledge of the user is taken from the reading records, and target knowledge is extracted from research ideas to establish the user's knowledge gap [13]. In most cases, a researcher will write a proposal that [18] outlines the objectives of a future study. For recommender systems, Zhang, Fuzheng, et al. [12] experimented with structural, textual, and visual material from the knowledge base [2]. In order to design CKE, they used embedding methods such as heterogeneous network embedding and deep learning embedding that extracts semantic representations from the sources like knowledge bases automatically [2]. It may also get representations of the feature from the knowledge bases simultaneously capturing the implicit relationship between users and items [2].

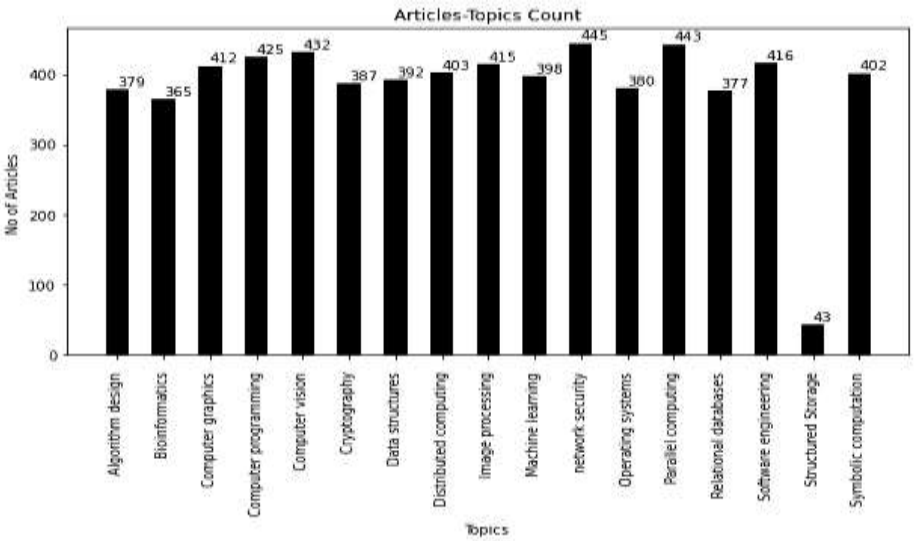

Fig. 1. Topic - Article Count

Dataset

A dataset of 46985 articles from Kowsari et.al was used for this study [2]. The articles relevant to computer science were selected from these articles. There are 6514 articles relating to computer science, organized into 17 different topics. A list of all papers in each cluster is shown in Fig. 1. There are attributes such as paper ID, Keywords, Abstract, and area within this dataset. In order to correctly recommend an article to the reader, the keywords and the abstract of the article are considered

Methodology

The architecture of this article is depicted in Figure 2. The purpose in this context is to find articles that are relevant to the search query. Prior to processing the user query, the similarity matrix must be generated. Cosine similarity is used to create the similarity matrix. It's used to see how similar the terms in the query are to the words in the target articles for comparison. The importance of each word in a document should be assessed before constructing the similarity matrix. To determine the relevance of words in a document, the TFIDF technique is utilized. The stages below will walk you through the model's operation.

- It begins with a thorough examination of all of the articles in the database.
- After that, the article's keywords and abstract are translated into individual units of words.
- After then, preprocessing is required for these words.

The initial stage in the preprocessing phase is to find stop words such as is, was, the, here, there, and so on. Stop words are words that have been detected as having no or low significance, and they are so eliminated from further processing.

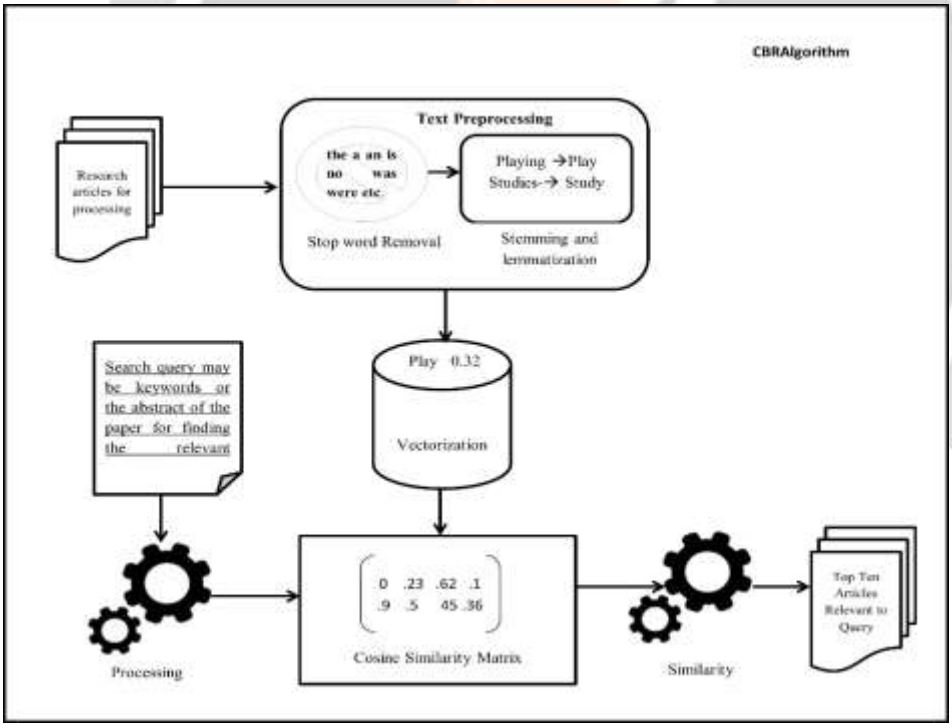

Fig. 2. Architecture

- Stemming and lemmatization are the next two steps in preprocessing. The root words are found using stemming, and the lemma of the words is discovered using lemmatization. The word studying, for example, has the root word "study" and studies have the lemma "study". In stemming, it tries to discover the word's suffix and prefix and eliminates them using basic stemming algorithms. However, when it comes to lemmatization, it attempts to find morphological words for the words that are given.
- Following the completion of the preprocessing, the result is fed into the vectorization process. The process of transforming extracted words to vectors by assigning a value to every word is known as

vectorization. The term frequency-inverse term frequency approaches are used to calculate the corresponding vectors.

- o the number of occurrences of the word in a document is computed using the term frequency.
- o the relevance with words in each the documents provided is estimated using the inverse document frequency method.
- o Finally, each word's TF-IDF is determined using these values.
- a similarity matrix is created after vectorization. The distance between words in a document is calculated using cosine similarity to create the similarity matrix.

The constructed model is ideal for recommending items to the user once it has been developed. The user enters a search query, which is subsequently processed by extracting the terms. The query's words are then used to calculate cosine similarity. Every document in the similarity matrix is matched with the query here. As a consequence, the algorithm retrieves the document with the highest cosine similarity scores [17] and recommends it to the user.

TF-IDF (Term Frequency-Inverse Document Frequency)

These metrics find how a word is relevant to the document. The Term frequency is used for calculating how a word is important in a document that contains the particular word. Inverse document frequency calculates how that particular word is common on the set of all other relevant documents. Equation (1) is used for calculating the term-frequency of a word in a document; it counts the log of frequency of the word in a particular document. In this article out of 17 different areas, only one area is considered as a document and the frequency of the word is searched in it. In equation (2) inverse-term-frequency is calculated by counting the word in every document that appears in the list of documents. In this article, the word is searched in all 17 clusters and counted if it is present in it. After calculating the term-frequency and inverse-term-frequency separately it is applied in equation (3) to calculate the TFIDF value, where is N is total number clusters.

$$\mathrm{TF}(\mathrm{word}, \mathrm{document}) = \mathrm{Log}\ (1+ \mathrm{frequency}\ (\mathrm{word}, \mathrm{document})) \quad (1)$$

$$\mathrm{IDF}(\mathrm{word}, \mathrm{document}_i \in \mathrm{Document}) = \log\left(\frac{N}{\mathrm{Count}(\mathrm{word}, \mathrm{document}_i \in \mathrm{Document})}\right) \quad (2)$$

$$\mathrm{TFIDF} = \mathrm{TF}(\mathrm{word}, \mathrm{document}) \cdot \mathrm{IDF}(\mathrm{word}, \mathrm{document}_i \in \mathrm{Document}) \quad (3)$$

Cosine similarity

Cosine similarity is used to measure how similar the given vectors are. It is calculated by dividing the dot products of vectors in the document by the magnitude of the vectors in the document as shown in equation (4). The vectors are calculated using TF-IDF and then the cosine-similarity is calculated to form a cosine matrix from the equation (4).

$$\mathrm{Cosine\ Similarity}\ (\mathrm{document1}, \mathrm{document2}) = \frac{\mathrm{Vectors}\ (\mathrm{document1}) . \mathrm{Vectors}\ (\mathrm{document2})}{|\mathrm{Vectors}\ (\mathrm{document1})| \quad |\mathrm{Vectors}\ (\mathrm{document2})|} \quad (4)$$

CBRAlgorithm

Definition 1: The dataset contains 17 different clusters that have several numbers of articles that are related with each other and these clusters are represented as C_i, Query from the user Q_i, CP_i is the preprocessed string of the respected clusters and the output of the algorithm is list of top 10 articles, A_i.

Algorithm 1: The Content based recommendation algorithm (CBRAlgorithm) takes input clusters C, and query Q from the user as the inputs. The algorithm starts its task by selecting the relevant cluster among the provided $C_{i,}$ cluster. For selecting the cluster that is relevant to the query this algorithm makes use of the preprocessed clusters, CP_i and query then it compares each of the clusters with the query Q by computing the similarity score. After calculating the similarity score it selects the cluster CP_i with high score and moves to next part of the algorithm.

The second of the algorithm attempts to list all the articles in the cluster that are similar the Q. Form these listed articles it chooses the top 10 articles with high similarity score. Here the algorithm progress by computing the vectors for each article, with which it computes the similarity score using cosine similarity. For vectorization it uses the methods Term frequency and inverse term frequency. It attempts to compare the vectors of each document with the user query Q. Finally the articles with high similarity score are considered relevant and those articles, A are returned as results.

Algorithm 1 CBRAlgorithm
Input Document Clusters : C Query : Q Preprocessed Clusters : CP N=10
Output List of top N articles: A
for article in C do Cp= Preprocess(article)
for word in Q do Preprocess(word)
for words in CP do generateVector(words)
for qwords in Q do GenerateVector(qwords)

maxclust=cp1
csim=0
for each cluster, ic in CP do
clist =compareSimScore(ic , Q)
if clist greaterthan csim do
maxclust = ic

for article in maxclust do GSC=ComputeSimScore(i , Q)
SortScore(GSC) count=0

for i in GSC do
if count less than N+1 do
assign i in A
increment count

for article in A do print article

Results and Analysis

For the search query "*Lightweight cryptography, IoT, Block cipher*" the result obtained by applying the methods TF-IDF and Cosine similarity is presented in Table 1. This method provides the top 10 computer science articles from the provided database that matches the search query. To evaluate the working of the model following metrics recall, precision, and F1 are used and the results are shown in Figure 3 [16]. Precision measures the percentage to which the model correctly recommends the articles that are relevant to the search query. Recall measures the percentage at which the relevant articles are recommended by the system to the total number of

articles that are relevant to the query provided by the user. F1 score combines both precision and recall for the evaluation of the model. Here the evaluation is done by considering the article that is recommended for the 10 best recommendations.

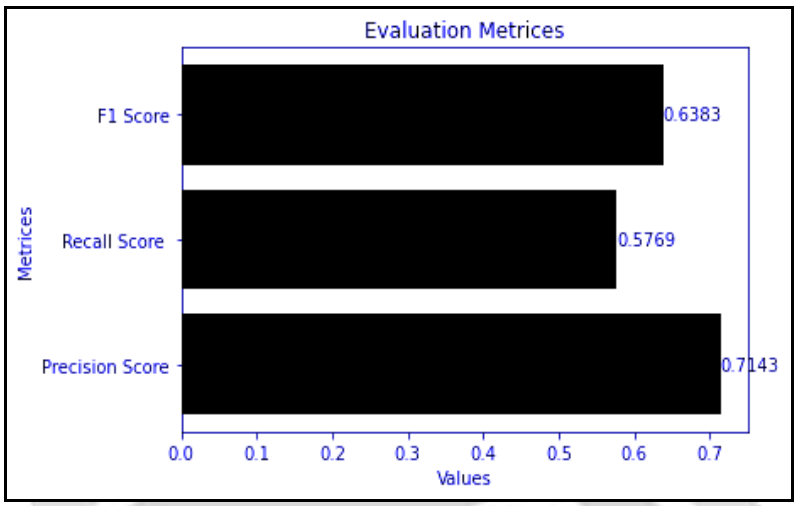

Fig. 3. Evaluation Metrics

Conclusion

The recommendation of an article to researcher from a vast research article data is made easier. It is also possible for researcher to fetch the interested article within considerably less amount of time than the traditional way of finding article.

Therefore, this article proposed an algorithm called CBRAlgorithm that is used to find the top 10 articles that matches the query of the user to find the relevant article. The result generated by the algorithm is then evaluated with the metrices like F1 score, Recall and precision score. Although the result not remarkable, we have planned to improve the algorithm with Co-Citation and Co-Author recommendations methods to address the problems of sparsity.

Fig. 4. Result for search Query *Lightweight cryptography*, *IoT*, *Block cipher*

References

1. K. Shah, A. Salunke, S. Dongare and K. Antala, "Recommender systems: An overview of different approaches to recommendations," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017, pp. 1-4, doi: 10.1109/ICIIECS.2017.8276172.
2. Dref1. Kowsari, Kamran; Brown, Donald; Heidarysafa, Mojtaba ; Jafari Meimandi, Kiana ; Gerber, Matthew; Barnes, Laura (2018), "Web of Science Dataset", Mendeley Data, V6, doi: 10.17632/9rw3vkcfy4.6. https://data.mendeley.com/datasets/9rw3vkcfy4/6
3. Li Y, Wang R, Nan G, Li D, Li M. A personalized paper recommendation method considering diverse user preferences. Decision Support Systems. 2021 Jul 1;146:113546.
4. Beel, J., Gipp, B., Langer, S. et al. Research-paper recommender systems: a literature survey. Int J Digit Libr 17, 305–338 (2016). https://doi.org/10.1007/s00799-015-0156-0
5. Xinyi Li, Yifan Chen, Benjamin Pettit, and Maarten De Rijke. 2019. Personalised Reranking of Paper Recommendations Using Paper Content and User Behavior. ACM Trans. Inf. Syst.37, 3, Article 31 (July 2019), 23 pages. DOI:https://doi.org/10.1145/3312528
6. K., Kan, MY. A comprehensive evaluation of scholarly paper recommendation using potential citation papers. Int J Digit Libr 16, 91–109 (2015). https://doi.org/10.1007/s00799-014-0122-2
7. Wang, G., He, X., & Ishuga, C.I. (2018). HAR-SI: A novel hybrid article recommendation approach integrating with social information in scientific social network. Knowl. Based Syst., 148, 85-99
8. Cai X, Han J, Li W, Zhang R, Pan S, Yang L. A Three-Layered Mutually Reinforced Model for Personalized Citation Recommendation. IEEE Trans Neural Netw Learn Syst. 2018 Dec;29(12):6026-6037. doi: 10.1109/TNNLS.2018.2817245. Epub 2018 Apr 12. PMID: 29993933.

9. Xia, Feng & Liu, Haifeng & Lee, Ivan & Cao, Longbing. (2016). Scientific Article Recommendation: Exploiting Common Author Relations and Historical Preferences. IEEE Transactions on Big Data. 2. 1-1. 10.1109/TBDATA.2016.2555318.
10. Son, Jieun, and Seoung Bum Kim. "Academic paper recommender system using multilevel simultaneous citation networks." Decision Support Systems 105 (2018): 24-33.
11. Manouselis, Nikos, and Katrien Verbert. "Layered evaluation of multi-criteria collaborative filtering for scientific paper recommendation." Procedia Computer Science 18 (2013): 1189-1197.
12. Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016, August). Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 353-362).
13. Weidong Zhao, Ran Wu, and Haitao Liu. 2016. Paper recommendation based on the knowledge gap between a researcher's background knowledge and research target. Inf. Process. Manage. 52, 5 (Sep 2016), 976–988. https://doi.org/10.1016/j.ipm.2016.04.004
14. Ma, X., Zhang, Y. & Zeng, J. Newly Published Scientific Papers Recommendation in Heterogeneous Information Networks. Mobile Netw Appl 24, 69–79 (2019). https://doi.org/10.1007/s11036-018-1133-9
15. Kowsari, K., Heidarysafa, M., Brown, D.E., Meimandi, K., & Barnes, L.E. (2018). RMDL: Random Multimodel Deep Learning for Classification. International Conference on Information System and Data Mining.
16. Sakib, N., Ahmad, R.B., Ahsan, M., Based, M.A., Haruna, K., Hiader, J., & Gurusamy, S. (2021). A Hybrid Personalized Scientific Paper Recommendation Approach Integrating Public Contextual Metadata. IEEE Access, 9, 83080-83091.
17. X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong and F. Xia, "Scientific Paper Recommendation: A Survey," in IEEE Access, vol. 7, pp. 9324-9339, 2019, doi: 10.1109/ACCESS.2018.2890388.
18. W. Zhao, R. Wu, W. Dai and Y. Dai, "Research Paper Recommendation Based on the Knowledge Gap," 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 2015, pp. 373-380, doi: 10.1109/ICDMW.2015.40.

Sivasankari.R, received bachelor degree in B.Sc computer science at Fatima College Madurai, 2011, and completed masters in MCA at Velammal college of Engineering and technology, Madurai, 2014. Currently doing a PhD in Computer Science. Specialization includes recommendation systems, semantic technologies, ontology, citation networks and IOT.
Email: manjurmoni@gmail.com

Dr.J.Dhilipan, Professor & Head, Department of MCA, SRMIST, Ramapuram. He completed M.Sc, MBA, MPhil and completed his Doctorate in 2014. His area of interest is Cloud Computing, E Commerce & Data Analysis. He published 30+ research articles in reputed journal and conference proceedings.
Email: hod.mca.rmp@srmist.edu.in