Hybrid scientific article recommendation system with COOT optimization

R. Sivasankari, J. Dhilipan

PII: S2666-7649(23)00051-6

DOI: https://doi.org/10.1016/j.dsm.2023.11.002

Reference: DSM 80

To appear in: *Data Science and Management*

Received Date: 18 April 2023

Revised Date: 2 November 2023

Accepted Date: 6 November 2023

Please cite this article as: Sivasankari, R., Dhilipan, J., Hybrid scientific article recommendation system with COOT optimization, *Data Science and Management* (2023), doi: https://doi.org/10.1016/j.dsm.2023.11.002.

Hybrid Scientific Article Recommendation System with COOT Optimization

Sivasankari.R [1], Dr.J.Dhilipan [2]
[1]Research Scholar, Department of Computer Applications(M.C.A),
SRM Institute of Science and Technology-Ramapuram Campus,
Ramapuram, Chennai, India
manjurmoni@gmail.com

[2]Professor and Head, Department of Computer Applications(M.C.A)
SRM Institute of Science and Technology- Ramapuram Campus,
Ramapuram, Chennai, India
hod.mca.rmp@srmist.edu.in

Hybrid scientific article recommendation system with COOT optimization

Abstract

Today, recommendation systems are everywhere, making a variety of activities considerably more manageable. These systems help users by personalizing their suggestions to their interests and needs. They can propose various goods, including music, courses, articles, agricultural products, fertilizers, books, movies, and foods. In the case of research articles, recommendation algorithms play an essential role in minimizing the time required for researchers to find relevant articles. Despite multiple challenges, these systems must solve serious issues such as the cold start problem, article privacy, and changing user interests. This research addresses these issues through the use of two techniques: hybrid recommendation systems and COOT optimization. To generate article recommendations, a hybrid recommendation system integrates features from content-based and graph-based recommendation systems. COOT optimization is used to optimize the results, inspired by the movement of water birds. The proposed method combines a graph-based recommendation system with COOT optimization to increase accuracy and reduce result inaccuracies. When compared to the baseline approaches described, the model provided in this study improves precision by 2.3%, recall by 1.6%, and MRR (Mean Reciprocal Rank) by 5.7%.

Keywords*: Recommendation system, COOT optimization, Citation network, Clustering, Long Short-Term Memory (LSTM)*

1. Introduction

Researchers face a significant challenge in identifying and searching for the most relevant papers for their work owing to the rapid growth of scientific knowledge and the increasing number of research articles published across numerous fields. To solve this issue, scientific article recommendation algorithms emerged as useful tools for supporting researchers in identifying the most relevant articles customized to their unique study objectives. These recommendation systems analyze and evaluate many accessible scientific literatures using modern algorithms and approaches, providing researchers with customized recommendations that fit their areas of interest.

Scientific article recommendation systems aim to improve the research process by automating the tasks of article discovery and searching, allowing researchers to focus on their primary jobs rather than wasting time searching for appropriate materials. To provide reliable and customized suggestions, these systems use various information sources, including metadata, article content, citation networks, and user preferences. These recommendation systems attempt to provide researchers with a customized and efficient experience by considering many aspects, such as content similarity, citation trends, and user feedback.

Content-based filtering is a common approach used in scientific article recommendation systems in which the system analyzes the textual content of articles and identifies similarities between articles based on keywords, topics, or other semantic factors. Another approach is collaborative filtering, which uses similar academics interests and behaviors to recommend articles that have been popular or frequently referenced within particular groups. Furthermore, hybrid recommendation systems incorporate various methodologies, such as content-based and collaborative filtering, to increase the quality and diversity of article recommendations.

Despite several recommendation systems, they encounter difficulties in addressing the following challenges.

（1） In content-based systems, the text of the articles plays a crucial role in making recommendations. However, obtaining the entire content of an article is often impractical; therefore, recommendations are often limited to only a portion of the article, such as the abstract. As a result, these algorithms face hurdles such as cold-start issues (when a new article lacks sufficient data to make an effective recommendation) and privacy concerns.

(2) Collaborative systems rely on author and paper citation networks for recommendations. However, they also face the cold-start problem when a cited paper is not listed in the database. Additionally, they may be affected by the "Mathew's effect," where cited articles are prioritized regardless of content similarity. Sparsity issues also arise because of the need to establish connections between large amounts of data, making the network construction complex.

(3) Hybrid systems combine content-based and collaborative filtering techniques to recommend articles. However, because hybrid systems are modeled as graph networks, special techniques are required to mitigate bias in the recommendation results.

This study proposes a Hybrid Scientific Article Recommendation System with COOT Optimization (HSARCO) to provide a curated list of relevant articles that aligns with a user's query. The suggested recommendation techniques utilize content-based filtering techniques to analyze article content and employ collaborative techniques to suggest articles based on citation relationships within the graph. To overcome the constraints typically observed in existing recommendation systems, the proposed hybrid graph recommendation method combines the capabilities of content-based and collaborative filtering strategies. This strategy aims to overcome challenges, such as the cold-start problem and sparse matrix issues, by incorporating a weighted semantic similarity index and COOT optimization, ultimately providing researchers with an efficient and accurate means of discovering relevant articles for their specific research needs.

One of the key limitations of existing systems is the cold-start problem, in which newly published or less-well-known articles lack sufficient data for effective recommendations. By incorporating a weighted semantic similarity index into the citation graph, the proposed strategy enhances content-based filtering by considering the semantic relationships between articles in the graph network. This index gives more weight to items that are more semantically comparable, boosting the accuracy and relevance of the recommendations. The system can better grasp the context and content of articles using powerful natural language processing techniques and semantic analysis, even when complete information is unavailable, thereby alleviating the cold-start problem.

Hybrid recommendation systems integrate content and graph-based approaches to successfully handle the cold-start problem. A content-based approach can provide initial recommendations whenever a researcher searches for a paper. The cold-start problem is avoided by using the text content of an article, which consists mainly of the abstract, title, and keywords. The citation graph adds further efficiency to the system by computing citations that match and recommend relevant papers. In addition to addressing the cold-start problem, the proposed hybrid graph recommendation strategy addresses the issue of sparse matrices. The vast amount of available scientific literature often leads to sparsity, making it challenging to establish meaningful connections and provide comprehensive recommendations. By integrating a graph-based recommendation system and COOT optimization, this strategy leverages the relationships between articles and citations to overcome sparsity issues. By constructing a robust graph network, the system can identify hidden relationships, uncover valuable connections, and provide diverse and accurate recommendations. The COOT optimization algorithm further enhance the performance of the recommendation system by efficiently optimizing the result-selection process, ensuring that the most relevant and high-quality articles were selected.

By incorporating the weighted semantic similarity index and COOT optimization into the hybrid graph recommendation strategy, researchers can benefit from an improved recommendation system that addresses the limitations of existing approaches. This system overcomes the challenges of the cold-start problem and sparse matrices, offering researchers an efficient and accurate means of discovering articles relevant to their research needs. By integrating advanced natural language processing techniques, semantic analysis, and graph-based recommendation systems, the proposed strategy empowers researchers to navigate the vast landscape of scientific literature confidently, thereby saving time and effort in searching for relevant articles. The main contributions of this study are as follows.

- The recommendation process begins with the construction of clusters, a critical phase in which the extensive collection of textual material is organized into coherent groupings, depending on their content. This clustering method is carried out using advanced techniques like Word2Vec vectorization and Long Short-Term Memory (LSTM) algorithms, which allow the system to discover detailed patterns and correlations within text data, laying the groundwork for accurate and successful article recommendations.
- Subsequently, a citation graph was created to depict the links between articles. Using this graph, the influential nodes can be determined using the inward and outward influential values.
- Finally, COOT optimization is used to select and compute articles that closely match the user's search query, ensuring that the recommended articles are highly relevant and customized to the user's needs.
- The efficiency of the algorithm was measured using precision, recall, and mean reciprocal rank metrics.

The remainder of this article is organized as follows: Section 2 presents a literature review, Section 3 describes the technique, Section 4 describes the experimental setup, Section 5 discusses the results and discussions, and Section 6 presents the conclusions.

2. Literature survey

A literature study for scientific article recommendation systems was conducted by classifying the articles into three primary categories: content-based, collaborative systems, and hybrid systems. The primary issues addressed to improve the performance of recommendation systems are cold-start, data sparsity, privacy, scalability, and diversity (Fayyaz et al., 2020). The following literature discusses the attempts to solve these issues using various methodologies.

Many content-based (CB) recommendation systems have been proposed, where the features used are derived from the text in the abstract. Wang et al. (2018) proposed CB model uses chi-square for feature selection and softmax regression to recommend the top one or the top three computer science-related articles. In the article proposed by Achakulvisut et al. (2016), the CB model utilizes latent semantic analysis (LSA) with the Rocchio algorithm and nearest neighbor assignment to recommend articles. In an article proposed by Rutkowski et al., 2018, the authors implemented the method that used a neuro-fuzzy approach for recommendation in CB models (Rutkowski et al., 2018). The method proposed for knowledge infusion into content-based recommender systems (KICBRS) uses knowledge infusion (KI), which contains word relationships for recommendations (Semeraro et al., 2009). The Content-Based Citation Recommendation (CB-CR) model uses a neural network and stochastic gradient descent (Bhagavatula et al., 2018). To address data sparsity and cold-start issues, Liu et al. (2021) suggested a triple cross-domain collaborative filtering (TCD-CF) (Liu et al., 2021). TCD-CF employs a transfer learning technique to generate intra-and inter-type information in both the auxiliary and target domains. The TCD-CF algorithm makes suggestions based on triplet auxiliary information such as the user, item, and rating-side relations with the target domain.

In citation recommendation, collaborative filtering can be used to suggest papers relevant to authors based on citation behavior. In the work implemented by Liu et al. (2015), reference papers were recommended to users based on a two-level citation relationship between the citing and cited papers. Wang and Blei (2011) attempted to merge a topic modeling technique with collaborative filtering for article recommendation. In Collaborative filtering with network representation learning for citation recommendation (CNCRecP) (Wang et al., 2020), the collaborative filtering method is combined with network embedding by constructing a rating matrix for the text information of a paper. The method implemented by Haruna et al. (2017) constructs paper-citation relations by collecting contextual metadata to recommend an article. The Collaborative denoising auto-encoder (CDAE) model (Wu et al., 2016) uses a denoising auto-encoder to construct a preference set that stores the binary data about the item that represents whether to recommend the item. The CDL (Wang et al., 2015) is a hierarchical Bayesian model that uses a sampling-based algorithm for recommendations.

Hybrid recommendation models combine content-based and collaborative filtering approaches to recommend articles. In the article proposed by (Sakib et al., 2021), an approach was taken in which the article's features, such as metadata, were used along with the paper-citation relationship. Hybridcite (Färber and Sampath,

2020) recommends citations using a semi-genetic hybrid recommender that utilizes the components by embedding information retrieved from the dataset, such as MAG, arXiv, and ACL. Scienstein (Gipp et al., 2009) is the first hybrid system to recommend research papers that reflected components such as citation analysis, explicit ratings, implicit ratings, author analysis, and source analysis. An MA article (Kanakia et al., 2019) uses a combination of co-citation and content-based embedding approaches with the MAG corpus. The CNRN model (Waheed et al., 2019) recommends research papers using a multi-level citation network and author network by constructing an ego node for the paper's structural relationship. The PR-HNE (Ali et al., 2020) model captures the changing pattern of researchers' interests and the significance of relationships, such as author information, topical relevance, labeled information, and venue information, using six information graphs and a weighted probabilistic approach to effectively represent the semantic relationships among networks. Sharma et al. (2021) proposed a deep learning model semantic personalization recommendation system (SPRS) for recommending videos based on their preferences and likes (Sharma et al., 2021). SPRS computes the rating and generates similarities based on user history and content behavior. The deep-learning model learns from previous searches and recommends the behavior of the user's desired film. In hybrid recommendation systems, it is difficult to determine the sequence for integrating two or more models and quantifying the weight of each model. To address this issue, Zhang et al. (2021) suggested a hybrid recommendation system that employs paragraph embeddings for user reviews and item descriptions. This model was used to detect the sentiments of reviews and attributes of objects in the Amazon product dataset (Zhang et al., 2021).

3. Methodology

HSARCO integrates context-based and collaborative filtering approaches with graph-based methods to address issues such as cold start and sparsity. A COOT optimization algorithm is proposed to address the challenge of finding the optimal solution within a large search space, which can be time-consuming with a substantial number of nodes and edges. The text data of the articles, including title, keywords, and abstract, were clustered using content-based filtering. In addition, a citation graph was constructed to facilitate the identification of relevant articles based on paper citations. The COOT algorithm is employed to determine the optimal values for computing the citation graph. Finally, the weights of the results were combined to provide recommendations for relevant articles. Fig. 1 shows the architecture of the proposed model.

The HSARCO system operates as follows. Initially, the dataset is clustered into thirteen different categories to streamline the computation of search queries and eliminate irrelevant content. When a user enters a search query, all clusters are matched to identify the most relevant cluster. The content-based (CB) approach, utilizing word2vec and LSTM techniques, was then employed to select the top N articles that best matched the search query within the selected cluster. The citation relationships of these articles were utilized to construct a graph. The citation graph was constructed by gathering paper-reference relationships from the top N articles. The next step involved assigning weights to these relations, considering the similarity between each article's relations in both graphs. COOT optimization was employed to select the top N papers from each graph. Finally, the weighted similarity scores obtained from the CB and citation graphs are used to recommend the final list of relevant articles.

3.1. Citation context clustering

Pre-processing these unstructured documents to structure them is required to cluster the text documents. Tokenization, stop-word removal, word embedding, and data pre-processing are some of the phases constituting this process. The process of cleaning and removing superfluous content from the provided data is known as data pre-processing. Tokenization divides the enormous text content of a document into smaller tokens. Lemmatization and stemming were then used to identify the patterns in these tokens. Word embeddings are produced at the end of data pre-processing after each token is checked to ensure that all stop words, such as pointless patterns, words, punctuation, and conjunctions, have been removed. Applying word embedding, each token is represented by a real-number vector computed using word2vect. These documents were clustered together.

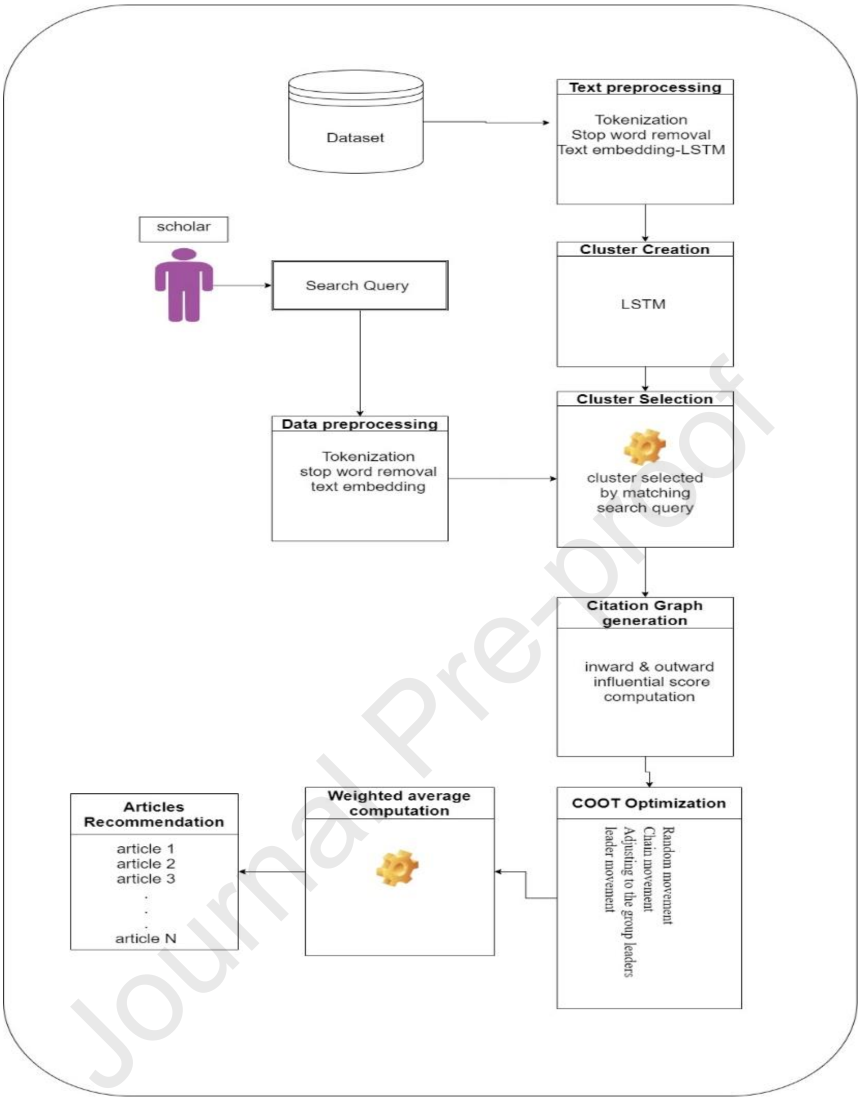

Fig.1 HSARCO Architecture Diagram.

Recurrent neural networks (RNN) are the most broadly applied neural network techniques owing to their effectiveness in processing sequential data. Despite its efficiency, RNN has two significant flaws: vanishing gradient and exploding gradient problems. Training a lengthy sequence of dependent text data is challenging because of these two limitations. The LSTM used the forget gate to address shortcomings of the RNN. The Forget gate maintains a record of the information to be retained, allowing a long sequence of text data to be processed in LSTM (Yu et al., 2019). This study used word2vec and LSTM to cluster related documents.

3.1.1. Word2Vec

Word2Vec converts words into vectors based on various characteristics, including the window size and vector dimensions (Jatnika et al., 2019). Word2vec uses a fixed-size vector to represent words. This vector efficiently captures semantic and syntactic similarities (Orkphol and Yang, 2019). It is modeled with one layer of an artificial neural network that predicts the value using a larger corpus of training text data. There are two variants of word2vec, namely CBOW and skip-gram. CBOW predicts the current word based on a list of words, and skip-gram predicts a list of words around the current word (Jatnika et al., 2019). Word2vec prevents overfitting by transforming the one-hot encoder values into continuous low-dimensional values. With this model, it is also possible to increase the efficiency by using a reduced training sample (Xiao et al., 2018).

3.1.2. LSTM

Recurrent neural networks are variants of artificial neural networks that can predict values based on their previous values and are reliable for processing sequential data. However, it also suffers from the vanishing gradient problem, where, during backpropagation, the gradient value of the neuron becomes very low or, in some extreme cases, zero. When this occurs, the neurons with zero gradients are not used for further processing. This problem is overcome in Long-short-term memory (LSTM) by introducing a memory cell into the hidden layer (Li et al., 2020). LSTM has three types of gates: an input gate, a forget gate, and an output gate. The input gate is used to determine the amount of information obtained from the current state of the cell. The forget gate determines how much information from the previous state should be remembered. The output gate determines the amount of output to be passed to the output of the cell (Li et al., 2020).

3.2. Citation graph

Definition1: Citation Graph G_{CG} is defined as G_{CG} = {N, E}, where N represents the set of nodes N = {Np,Nr }, and E represents the set of edges from the graph E={ Er, Ec}. In the node-set, N is the citation graph $G_{CG, Np}$ represents the paper node, and Nr is the reference node. In the set of edges E, Ec denotes the citing edge, and Er denotes the co-cited edge.

The citation graph shows relevant articles responding to user queries (Zhang and Ma, 2020). In the context of information retrieval and recommendation systems, citation graphs have evolved into strong tools for proposing appropriate articles in response to user queries. The intrinsic benefit of citation graphs is their capacity to include the most recent and relevant research articles, ensuring readers can access the most recent academic literature. A citation graph is composed of a collection of nodes, each representing a research article, and a network of edges that indicate the citation links between these papers. We considered two articles, A and B, to demonstrate this notion. Article A cites Papers X, Y, and Z, whereas Article B cites Papers X, Y, and W. We notice an intriguing prospect here: Article B may be recommended concerning a user’s inquiry. The justification for this proposal stems from the fact that articles B and A have two or more citations, implying a thematic or intellectual resemblance between them. However, fundamental condition must be met for this advice to be valid: the text-similarity of articles A and B must exceed a preset level, showing the significant overlap in their substances and subject matter.

In the citation graph, articles are represented as nodes, whereas the citing connections between them are represented as edges. Each node in this graph has two unique values: the inward and outward influential. The inward influential value indicates how frequently other articles in a network refer to a certain article. By contrast, the outward influential value represents the total number of articles cited. These dual values provide a full perspective on an article’s influence within the citation graph, including both its impact on others by being frequently referenced and its contribution through citations to other articles.

3.3. COOT optimization

COOT optimization is a meta-heuristic optimization (Naruei and Keynia, 2021) technique that uses the behavior of small water birds to determine the optimum solution (Malavath and Devarakonda, 2022; Naruei and Keynia, 2021). The behavior of this bird on the surface of the water is composed of four types of movement to search for food, as follows:

- Random movement

- Chain movement
- Adjusting the position based on the group leaders
- Leader movement

COOT algorithm

The COOT algorithm is used to compute the position of each component i at each iteration and update it based on its movements. It begins by initializing the values in a random population using the Eq.(1)

$$Cootpos(i) = rand(1, d) * (ub - lb) + lb \tag{1}$$

where CootPos(i) represents the position of the coot; d is the number of variables or dimensions of the problem; and ub and lb represent the upper and lower bounds of the search space, respectively (Naruei and Keynia, 2021).

Random movement computation: In random movements, coot birds move randomly on their sides (Qin et al., 2022). It attempts to find its position towards the food by randomly positioning itself in the swarm. In this algorithm, random movement updates the value of i randomly to avoid problems with the local optimum using Eq. (2), where R_2 is a value between [0,1] inclusive, and the value of A is computed using Eq. (3).

$$Cootpos(newi) = Cootpos(i) + A * R2 * (Cootpos(i) - Cootpos(initial)) \tag{2}$$

$$A = 1 - iter * (\frac{1}{MaxIter}) \tag{3}$$

Chain movement: Chain movement helps each coot adjust its position within the flock. This is implemented in the algorithm by calculating the average distance between two coots using the Eq. (4).

$$Cootpos(i) = 0.5 * \left(Cootpos(i-1) + Cootpos(i)\right) \tag{4}$$

Adjusting the position based on the group leaders: In each flock of coots, there are two or more groups, each led by a leader. The members of this small group adjust their positions according to the position of the group leader. To implement this, the algorithm chooses one local optimum value as the leader of the group, and the other member positions are adjusted accordingly. It is calculated using the Eq. (5), where Cootpos(i) shows the current position of the coot, LeaderPos(k) denotes the selected leaders' position, R1 means a random number between 0 and 1, and R denotes a random number between −1 and 1(Qin et al., 2022).

$$Cootpos(i) = Leadrpos(K) + 2 * R1 * COS(2R\pi) * (Leadrpos(K) - Cootpos(i)) \tag{5}$$

Group leader K is selected by each member using Eq. (6), where K represents the index of the leader, i represents the total number of coot bird followers, and NL is the total number of leaders.

$$K = 1 + (i\ MOD\ NL) \tag{6}$$

Leader movement: The positions of the group leaders were adjusted by the flock leader. The local optimum values were updated following the global optimum value using Eq. (7).

$$Leadrpos(i) = \begin{cases} B * R3 * cos(2R\pi) * \left(gBest - LeaderPos(i)\right) + gBest & R4 < 0.5 \\ B * R3 * cos(2R\pi) * \left(gBest - LeaderPos(i)\right) - gBest & R4 \geq 0.5 \end{cases} \tag{7}$$

where gBest is the optimal location of every iteration, random numbers R3 and R4 take values between 0 and 1, random number R is in the interval [−1,1], and B should be found based on Eq. (8).

$$B = 2 - L * \left(\frac{1}{maxiter} \right) \tag{8}$$

4. Experimental setup

This study used the citation network dataset of Tang et al., where the citations were taken from various sources, including DBLP, ACM, MAG, and other comparable sources (Tang et al., 2008). The citation research for this study used the DBLP citation network, version 13. Nearly 5,354,309 distinct publications with approximately 48,227,950 citation relationships constitute this version of the DBLP. The dataset was broadly clustered into 13 main groups: computer science (CS), Electronics and Communications Engineering (ECE), Mechanical and Automation Engineering (MAE), and civil engineering. The primary features of the DBLP citation network dataset version 13 include the title, keywords, abstract, field of study, references, author, venue, and year of publication.

The dataset is initially pre-processed to extract the text, which undergoes tokenization, stop-word removal, word embedding, and data pre-processing. The proposed method uses LSTM with word2vec embedding to cluster articles and trains them with the number of units in the LSTM layer set to 128 and 100 epochs, a learning rate of 0.00005, and an adaptive moment estimation optimizer for optimization. The mean-squared error (MSE) was used as the objective function for training. A dropout rate of 0.2 is applied to the LSTM layer, which randomly dropped 20% of the units to prevent overfitting. L2 regularization was applied with a lambda value of 0.01 to control overfitting. The dimensions of the word embedding were set to 300. The activation function used rectified linear units and softmax, and the cluster size of the dataset was set to 15. Table 1 lists the parameters used to train LSTM for the proposed model.

Table 1. Training parameters for the proposed approach

Parameter	**Values**
LSTM Layer Units	128
Number of Epochs	100
Learning Rate	0.00005
Optimizer	Adaptive Moment Estimation (Adam)
Objective Function	Mean Squared Error (MSE)
Dropout Rate	0.2 (20% of units dropped)
L2 Regularization Lambda	0.01
Word Embedding Dimension	300
Activation Functions	Rectified Linear Units (ReLU), SoftMax
Cluster Size	15

The next step after the clustering of the dataset was the construction of the citation graph. For the optimization of the graph, the COOT algorithm was used. The weights of the articles were computed using a citation graph. Finally, the results were calculated by computing the weighted average of all papers using Eq. (9).

$$\text{WeightA (i)} = \frac{\text{art(i)} + \text{cite(i)}}{2} \tag{9}$$

where i is an article that belongs to the Artilist, art(i) represents the text similarity score of article i computed using LSTM with word2vec, cite(i) represents the influential score of article i computed using a citation graph with COOT optimization, and weightA(i) represents the average weight computed for article i.

Algorithm 1 HSARCO

```
Input: List of research articles $A=\{a_1,a_2,a_3,.....,a_n\}$in and query string $Q=\{q_1,q_2,q_3,.....,q_n\}$
Output: List of recommended research articles $RA=\{rd_1,rd_2,rd_3,.....,rd_n\}$ that are relevant to the search query
#Extraction
For each article in A do
    Obtain PT title of each paper and PAb abstract of each paper
    $PT=\{t_1, t_2, t_3,.....,t_n\}$ and $PAb=\{ab_1, ab_2, ab_3,.....,ab_n\}$
    Obtain $PR_i$ reference of each paper $PRi = \{pr_1, pr_2, pr_3,.....,pr_n\}$
End For
#Preprocessing
For each PT and PAb do
    Generate PW list of word tokens obtained after Preprocessing PT and PAb
    $PW_i=\{NLTK(PT)+NLTK(PAb)\}$
End For
#Clustering
For PW do
    Compute PC list of Clusters with similar concept
    $PC_i =\{LSTM (word2vec(PW_i))\}$
End For
#Choosing Relevant Cluster
For each $PC_i$ do
    Compute $SimScore_i$ of each cluster in $PC_i$ with Q, $PC_i= \{c_1, c_2, c_3,.....,c_n\}$
    Choose $PC_i$ with largest $SimScore_i$
    $SPC=\{c_i$ with largest SimScore$\}$
End For
For each paper SPC $_{do}$
    Compute CScore by comparing with Q
    $CScore_i$ = Similarity (spci, Q)
    Generate PCN, List of top N papers with high Similarity
    $PCN=\{spc_i$ with largest CScore $\}$
End For
For list of papers in PCN do
    Construct the graph $G_{CG}$
    Compute CGS score using COOT technique
    Generate CGN, List of top N papers in citation graph
End For
For each paper in PCN, CGN and DO
    Compute RA weight of each paper
    $RA_i$ = Weight $(PCN_i + CGN_i\}$
End For
Result: List top N papers in $RA_i$
```

Algorithm1: The HSARCO algorithm processes the input by obtaining the articles from the DBLP v13 dataset in JavaScript Object Notation (JSON) format and then extracts the required data items, such as paperID, title, abstract, and references, from the dataset. It then proceeds with data pre-processing, in which the text data from the titles and abstracts of the articles are treated for further processing. Then, these data were again processed using techniques such as tokenization, stop-word removal, stemming, lemmatizing, and word embedding using

the Natural Language Toolkit (NLTK). Using word2vec and LSTM, similar articles were clustered, and the query string was compared with each cluster to find the relevant cluster that matched the query. After finding a cluster with a high similarity score, all articles in the chosen clusters were compared with a user query to find the top N matches. In the next stage of the process, a citation graph was created. Subsequently the top N papers with high similarity were computed from the citation graph using the COOT optimization technique.

The basic purpose of COOT optimization is to identify highly important articles within a particular citation network. This technique includes many fundamental tactics, such as random exploration, chain exploration, influence adjustment, and prioritization of highly cited publications. This procedure involved rating each node in the network to assess its influence by considering both inward and outward citations. The citation graph was used as the input, and the result was a ranked list of articles based on influential scores, which were then supplemented with text similarity scores for user query matching. Starting with a randomly selected collection of articles, the optimization process employs COOT techniques to carefully examine the citation graph and iteratively improve the selection of influential articles. This method seeks to improve the quality of suggestions and search results by selecting the articles with the greatest citation network influence. This procedure involved rating each node in the network to assess its influence by considering both inward and outward citations. Finally, the overall weights of the articles were computed using the similarity score of each document with the influential score generated in the citation graph. The result of the algorithm is a list of the top N articles that match the user's query and are suggested by the proposed algorithm.

The computational complexity of HSARCO depends on the number of iterations, citation graph size, cluster size, and number of articles in each cluster. The time complexity for the initial clustering process is O(K), where K is the number of clusters. To compute the text similarity score, the complexity is O(N), where N is the total number of articles in the clusters. For citation graph generation, the complexity directly depends on the number of edges E and the number of nodes D; therefore, the complexity is O (E + D). Finally, the complexity of the citation graph exploration using COOT optimization is O (G*(E + D)), where G is the total number of local group leaders. Based on the time complexity of each step in the proposed HSARCO algorithm, the factors that directly affect the complexity are the number of nodes D, the number of Edges E, and the group population of COOT G. The overall complexity is determined by the most time-consuming step, which is citation graph exploration using COOT optimization O (G*(E + D)). Therefore, the total computational complexity of this algorithm is O (G*(E + D)).

Adapted COOT-optimization approach:

The main objective of the COOT optimization algorithm in this study is to identify highly influential articles in the citation graph by efficiently exploring the graph. The COOT optimization approach is used to explore the entire graph in a balanced manner by visiting every node in the graph to discover influential articles in the graph by randomly performing exploration, chaining, prioritizing the highly influential nodes in the graph by choosing the node, and selecting nodes by considering the leader.

In COOT, during the exploration process, the score of each node is updated using an algorithm upon visiting each node. Every time a node is visited based on a high-citation score node (leader) from the entire graph, nodes with high citations (group leaders) are constantly generated. These processes are repeated until the stop condition for the algorithm is satisfied, and the leader nodes (both leader and group leaders) continue to change. Upon reaching the stop condition of an algorithm, where no nodes are left to visit, the algorithm stops and finalizes the highly influential nodes. With COOT, it is very simple to identify the most influential nodes, as they have one leader node and N local leader nodes, along with some mildly influential nodes from the local groups. All these nodes were selected for weighted average computation to generate the recommendation list.

Terms in COOT optimization:

（1） *Random exploration (Random movement):*

Select a random article within the citation graph.

Explore its citations.

(2) *Chain exploration (Chain movement):*
Choose a highly cited article as the starting point.
Follow its citation chain to explore a sequence of influential articles.

(3) *Influence adjustment (Adjusting position based on group leaders):*
Identify papers that have a high number of citations within the field
Adjust the exploration focus based on the connections to these influential articles.

(4) *Prioritizing highly cited papers (Leader movement):*
Prioritize articles that are highly cited within the citation graph.
Explore articles that are cited by many others.

(5) *Score:*
To calculate the influential score of each node in the graph, the total score was computed by summing each node's inward and outward scores.

(6) *Input:*
A citation graph is provided as the input.

(7) *Output:*
In the case of the citation graph, the output of this process was a list of articles ranked with high influential scores. These articles were provided with text similarity scores by comparing the search queries of the user.

Steps in the optimization process:

(1) Start with an initial set of articles randomly.
(2) Apply the adapted COOT-optimization strategies to explore the citation graph.
(3) Calculate Influence Scores for each article based on its citation patterns.
(4) Iterate through the exploration process to refine the set of influential articles.

5. Result and discussions

The results were evaluated using three metrics: precision, recall, and Mean Reciprocal Rank (MRR). Precision measures the effectiveness of the recommendations by considering the actual list of recommendations and the recommendations made by the system using Eq. (10). Recall measures the proportion of positively relevant articles in the top N recommendation lists using Eq. (11). The MRR is the mean reciprocal rank that evaluates the system by concentrating the reciprocal relative rank of each item i in the top N list of articles, as shown in Eq. (12).

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True positive} + \text{False positive})} \tag{10}$$

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True positive} + \text{False Negative})} \tag{11}$$

$$MRR = \frac{1}{N}\sum_{i=1}^{N} Rank_i \tag{12}$$

The terms true positive, false positive, and false negative from Eq. (10) and (11) are defined as follows:

- True positive: articles listed in the top N list and recommended by the proposed model in the top N list.
- False positive: False positive articles are not listed in the top N list but are recommended by the proposed model in its top N list.
- False negative: articles that are not listed in the top N list and also not recommended by the proposed model in its top N list.

Table 2 summarizes the results obtained by evaluating the proposed method. The evaluation was based on the computation of precision, recall, and Mean Reciprocal Rank (MRR) scores, which were used to assess the performance of the algorithm. The proposed HSARCO algorithm incorporates four levels of working approaches, and the evaluation was performed at different values of k (20, 50, and 100) to analyze the results.

Table 2 Performance evaluation of the used approaches.

			Recall		
Approach	**Precision**	**MRR**	**@20**	**@50**	**@100**
word2Vec + LSTM	0.0534	0.313	0.434	0.515	0.711
word2Vec + LSTM + citation graph	0.0791	0.347	0.448	0.635	0.734
word2Vec + LSTM + citation graph + COOT optimization	0.1621	0.6944	0.599	0.720	0.7629

In the first level of the approach, a straightforward content-based recommendation system was developed using Word2Vec and LSTM. This initial approach focused on utilizing the titles and abstracts of the articles as the primary attributes for recommendations. However, the accuracy of this approach is not particularly high. To improve accuracy, the approach was further enhanced with graph-based recommendations. This involves the utilization of citations to recommend articles. By incorporating these graph-based recommendations, the results showed an increase in accuracy compared to the initial approach.

Furthermore, the algorithm was upgraded by introducing COOT optimization techniques. This upgrade significantly improves the results obtained using the recommendation system. It was observed that combining the methods of Word2Vec, LSTM, citation graph, and COOT optimization led to the best results in all scenarios, resulting in a significant increase in accuracy. The analysis of the proposed algorithm demonstrates that it achieves the best results when the Word2Vec, LSTM, citation graph, and COOT optimization methods are combined. This comprehensive approach substantially increases the accuracy of the article recommendation system.

Table 3 Performance comparison of baseline models with proposed model.

Models	**Precision**	**Recall**	**MRR**
Proposed method	0.1621	0.7629	0.6944
RSHK-RNN	0.1590	0.4055	0.6568
CB-CR	0.1558	0.5329	0.5481
ClusCite	0.0446	0.7502	0.3193
DocCit2Vec	0.0216	0.6786	0.2616

The proposed model was compared with baseline models, and the results are summarized in Table 3.

Baseline methods:

ClusCit (Ren et al., 2014): Cluscite attempts to cluster the citation based on the relevance of the query in the heterogeneous bibliographic network. After grouping them based on relevance, it finds the group with the utmost relevance to the query, and based on that, it recommends the document.

CB-CR (Bhagavatula et al., 2018): This study proposes a model trained using a neural network and stochastic gradient descent to recommend global citations. It is suitable for recommending citations even with vast amounts of data. In addition, this model overcomes the missing metadata values by concentrating only on the text content of the articles.

DocCit2Vec (Zhang and Ma, 2020): In this study, the authors used the architectures of two different neural networks, one with a conventional average hidden layer and the other with an attention hidden layer, to recommend articles based on the structural context of the document.

RSHK-RNN (Zhu et al., 2021): In this study, the authors constructed bibliographic networks using the metadata of research articles by converting metadata relations into vectors using TransD and PV-DM models, which were used for text vectorization. The recommendation model uses an attentive bidirectional RNN that focuses on user identity.

In this study, the performance of the proposed HSARCO model was compared with those of four other models. This study aimed to evaluate the performance in terms of precision, recall, and Mean Reciprocal Rank (MRR). The CB-CR baseline model, which measures the accuracy of the recommended articles, had the highest precision. Compared with CB-CR, the proposed HSARCO model showed a significant improvement in precision of 2.3%. This suggests that the recommendations made by the HSARCO model are more precise. ClusCite outperformed the baseline models in terms of recall, which measures the capacity of a model to locate pertinent articles. However, the proposed HSARCO model, outperforms ClusCite recall by 1.6%, demonstrating its superior ability to find pertinent articles. The RSHK-RNN baseline model was compared to the Mean Reciprocal Rank (MRR) metric, which established the average rank of the first relevant article in the recommendation list. Compared with the RSHK-RNN, the proposed HSARCO model showed a remarkable improvement in MRR of 5.7%. This improvement suggests that the HSARCO model is more likely to place relevant articles on a list of suggested readings.

The evaluation results demonstrate that the proposed HSARCO model outperforms the four baseline models in terms of article recommendation. The HSARCO model outperformed the CB-CR model in terms of precision by 2.3%, ClusCite in terms of recall by 1.6%, and MRR by 5.7% when compared with the RSHK-RNN. According to these results, the HSARCO model is more precise, efficient, and reliable for recommending relevant articles.

6. Conclusions

The focus of this study is to provide a recommendation system that suggests a relevant list of articles based on user queries. The system considers the context of the articles, including the title, keywords, and abstract, as well as metadata such as citation relations. To achieve this, the proposed hybrid method incorporates techniques like LSTM and word2Vec for analyzing the contextual information of the articles and calculating article similarity indexes. Additionally, COOT optimization was used to select articles from the citation networks, ensuring that the most relevant and valuable articles were recommended. The evaluation results demonstrate the notable improvements achieved by the proposed system. Compared to the baseline methods discussed in this paper, the Hybrid Scientific Article Recommendation System showed enhancements of 2.3% precision,1.6% recall, and 5.7% MRR. These improvements indicate the ability of the system to provide more accurate and relevant article recommendations to users. By employing these evaluation metrics, this study demonstrated the effectiveness and performance gains of the system, solidifying its value in assisting researchers in finding articles relevant to their research needs.

References

Achakulvisut, T., Acuna, D. E., Ruangrong., et al., 2016. Science Concierge: A fast content-based recommendation system for scientific publications. PloS one. 11(7) (2016), p. e0158423.

Ali, Z., Qi, G., Muhammad, K., et al., 2020. Paper recommendation based on heterogeneous network embedding. Knowledge-Based Systems. 210 (12) (2020), p. 106438.

Bhagavatula, C., Feldman, S., Power, R., et al., 2018. Content-Based Citation Recommendation. In:2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 1, pp. 238-251.

Färber, M., Sampath, A., 2020. Hybridcite: A hybrid model for context-aware citation recommendation. In: 2020 ACM/IEEE joint conference on digital libraries. pp. 117-126.

Fayyaz, Z., Ebrahimian, M., Nawara., et al., 2020. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. Applied sciences.10(21) (2018), p. 7748.

Gazdar, A., Hidri, L., 2020. A new similarity measure for collaborative filtering-based recommender systems. Knowledge-Based Systems. 188 (1) (2020), p. 105058.

Gipp, B., Beel, J., Hentschel, C., 2009. Scienstein: A research paper recommender system. In: 2009 The international conference on Emerging trends in computing. pp. 309-315.

Haruna, K., Akmar Ismail, M., Damiasih, D., et al., 2017. A collaborative approach for research paper recommender system. PloS one. 12(10) (2017), p. e0184516.

Jatnika, D., Bijaksana, M. A., Suryani, A. A., 2019. Word2vec model analysis for semantic similarities in english words. In:2019 Procedia Comp Science. 157 (2019), pp. 160-167.

Kanakia, A., Shen, Z., Eide, D., et al., 2019. A scalable hybrid research paper recommender system for microsoft academic. In: 2019 The world wide web conference. pp. 2893-2899.

Li, S., Li, W., Cook, C., et al., 2018. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In:2018 IEEE conference on computer vision and pattern recognition. pp. 5457-5466.

Li, T., Hua, M., Wu, X. U., 2020. A hybrid CNN-LSTM model for forecasting particulate matter (PM2. 5). IEEE Access. 8 (Feb) (2018), pp. 26933-26940.

Liu, H., Kong, X., Bai, X., et al., 2015. Context-based collaborative filtering for citation recommendation. IEEE Access. 3 (Sep) (2015), pp. 1695-1703.

Liu, T., Deng, X., He, Z., et al., 2021. TCD-CF: Triple cross-domain collaborative filtering recommendation. Pattern Recognition Letters. 149 (Sep) (2021), pp. 185-192.

Malavath, P., Devarakonda, N., 2022. Coot optimization based Enhanced Global Pyramid Network for 3D hand pose estimation. Machine Learning: Sci and Tech. 3(4) (2022), p. 045019.

Manaswi, N.K., 2018. RNN and LSTM. Deep Learning with Applications Using Python. Apress, Berkeley, CA. pp.115-126.

Mooney, R. J., Roy, L., 2000. Content-based book recommending using learning for text categorization. In:2000 fifth ACM conference on Digital libraries. pp. 195-204.

Naruei, I., Keynia, F., 2021. A new optimization method based on COOT bird natural life model. Expert Systems with Applications. 183 (Nov) (2021), p. 115352.

Naruei, I., Keynia, F., 2022. Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. Engineering with computers. 38 (4) (2022), pp. 3025-3056.

Orkphol, K., Yang, W., 2019. Word sense disambiguation using cosine similarity collaborates with Word2vec and WordNet. Future Internet. 11(5) (2019), p. 114.

Pandey, A. C., Garg, M., Rajput, S., 2019. Enhancing text mining using deep learning models. In: 2019 Twelfth International Conference on Contemporary Computing. IEEE, pp. 1-5.

Qin, L., Xu, T., Li, S., et al., 2022. Coot algorithm for optimal carbon–energy combined flow of power grid with aluminum plants. Frontiers in Energy Research. 10 (2022), p. 856314.

Ren, X., Liu, J., Yu, X., et al., 2014. Cluscite: Effective citation recommendation by information network-based clustering. In:2014 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 821-830.

Rutkowski, T., Romanowski, J., Woldan, P., 2018. A content-based recommendation system using neuro-fuzzy approach. In: 2018 IEEE International Conference on Fuzzy Systems. IEEE, pp. 1-8.

Sakib, N., Ahmad, R. B., Ahsan, M., et al., 2021. A hybrid personalized scientific paper recommendation approach integrating public contextual metadata. IEEE Access. 9 (June) (2021), pp. 83080-83091.

Semeraro, G., Lops, P., Basile, P., et al., 2009. Knowledge infusion into content-based recommender systems. In: 2009 the third ACM conference on Recommender systems. pp. 301-304.

Sharma, S., Rana, V., Kumar, V., 2021. Deep learning based semantic personalized recommendation system. International Jour of Info Management Data Insights. 1(2) (2021), p. 100028.

Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena. 404 (Mar) (2020), p. 132306.

Tang, J., Zhang, J., Yao, L., et al., 2008. Arnetminer: extraction and mining of academic social networks. In:2008 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 990-998.

Waheed, W., Imran, M., Raza, B., et al., 2019. A hybrid approach toward research paper recommendation using centrality measures and author ranking. IEEE Access. 7 (Feb) (2019), pp. 33145-33158.

Wang, C., Blei, D. M., 2011. Collaborative topic modeling for recommending scientific articles. In:2011 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 448-456.

Wang, D., Liang, Y., Xu, D., et al., 2018. A content-based recommender system for computer science publications. Knowledge-Based Systems. 157 (Oct)(2018), pp. 1-9.

Wang, H., Wang, N., Yeung, D. Y., 2015. Collaborative deep learning for recommender systems. In:2015 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1235-1244.

Wang, W., Tang, T., Xia, F., et al., 2020. Collaborative filtering with network representation learning for citation recommendation. IEEE Transactions on Big Data. 8(5) (2020), pp. 1233-1246.

Wu, C., Wang, B., 2017. Extracting topics based on Word2Vec and improved Jaccard similarity coefficient. In: 2017 IEEE second international conference on data science in Cyberspace. IEEE, pp. 389-397.

Wu, Y., DuBois, C., Zheng, A. X., et al., 2016. Collaborative denoising auto-encoders for top-n recommender systems. In: 2016 the ninth ACM international conference on web search and data mining. pp. 153-162.

Xiao, L., Wang, G., Zuo, Y., 2018. Research on patent text classification based on word2vec and LSTM. In: 2018 11th International Symposium on Computational Intelligence and Design. IEEE, 1 (2018), pp. 71-74.

Yu, Y., Si, X., Hu, C., et al., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural computation. 31(7) (2019), pp. 1235-1270.

Zhang, Y., Ma, Q., 2020. Doccit2vec: Citation recommendation via embedding of content and structural contexts. IEEE Access. 8 (June) (2020), pp. 115865-115875.

Zhang, Y., Liu, Z., Sang, C., 2021. Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation. Applied Soft Computing. 106 (July) (2021), p. 107345.

Zhu, Y., Lin, Q., Lu, H., et al., 2021. Recommending scientific paper via heterogeneous knowledge embedding based attentive recurrent neural networks. Knowledge-Based Systems. 215 (Mar) (2021), p. 106744.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

NONE