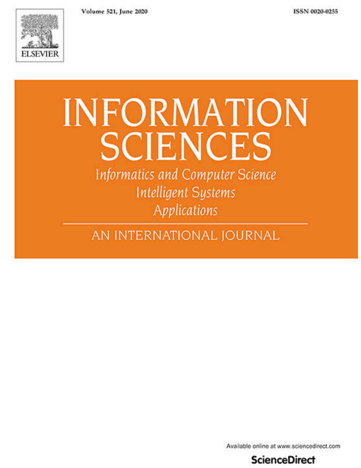
Journal Pre-proofs



SHARE: Designing Multiple Criteria-Based Personalized Research Paper Recommendation System

Arpita Chaudhuri, Monalisa Sarma, Debasis Samanta

PII: S0020-0255(22)01115-X
DOI: https://doi.org/10.1016/j.ins.2022.09.064
Reference: INS 18104

To appear in: *Information Sciences*

Received Date: 19 January 2022
Accepted Date: 26 September 2022

Please cite this article as: A. Chaudhuri, M. Sarma, D. Samanta, SHARE: Designing Multiple Criteria-Based Personalized Research Paper Recommendation System, *Information Sciences* (2022), doi: https://doi.org/10.1016/j.ins.2022.09.064

SHARE: Designing Multiple Criteria-Based Personalized Research Paper Recommendation System

Arpita Chaudhuri[a], Monalisa Sarma[b], Debasis Samanta[c*]

[a] *Advanced Technology Development Centre, Indian Institute of Technology Kharagpur, Kharagpur 721 302, India*

[b] *Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721 302, India*

[c] *Subir Chowdhury School of Quality and Reliability, Indian Institute of Technology Kharagpur, Kharagpur 721 302, India*

Abstract

Extraneous growth of scientific information over the Internet makes the searching task non-trivial and as a consequence researchers are facing difficulties in finding relevant papers from the millions of research papers in digital repositories. The research paper recommendation systems have been advocated to address this problem. The existing research paper recommendation systems lack in exploiting prominent information of papers, such as relevancy with the current time, novelty, scientific contribution, writing complexity of the papers, etc. Further, the existing models emphasize only on user's preference rather than user's intention that may change with time. Furthermore, the existing models do not consider a sound ranking strategy to unleash the personalization aspect and relevancy of papers. This work aims to address the existing limitations and proposes a systematic hidden attribute-based recommendation engine (SHARE). SHARE utilizes a feature engineering technique to unfold valuable insights of papers through multiple hidden features. These features are used as a context for users as well as multiple criteria for ranking papers. Additionally, SHARE predicts a user's intention beyond the user's preference to capture the dynamic notion of a user. Finally, a novel ranking strategy is proposed to retrieve personalized and the most important papers. SHARE is flexible for recommending both old and new users. In order to evaluate the effectiveness of SHARE both user studies and system evaluations were performed. Experimental results substantiate the efficacy of the proposed approach and are comparable to the existing systems.

Key words: Research paper recommendation, Content-based recommendation, User modeling, Multi-criteria analysis, Hybrid ranking, Personalized recommendation.

*Corresponding author
Email addresses: arpita.chaudhuri@iitkgp.ac.in (Arpita Chaudhuri[a]), monalisa@iitkgp.ac.in (Monalisa Sarma[b]), dsamanta@iitkgp.ac.in (Debasis Samanta[c*])

1. Introduction

The increasing growth of data on the Internet makes the searching task non-trivial. It is expected that worldwide data will reach 175 zettabytes in 2025 [1]. Considering this Big Data scenario, recommendation systems have been pioneered in different domains, such as food, social([1, 2]), music, etc. In recent years, due to the excessive growth of scientific articles on the Internet, researchers are facing difficulties in finding relevant papers. It was reported by the US National Science Foundation (NSF) that worldwide the number of scientific articles increased from 1,755,850 in 2008 to 2,555,959 in 2018 [2]. However, the system for a recommendation of research articles is scarcely reported.

The paper recommendation systems can be broadly classified into three categories: (1) Content-based filtering (CBF) [3, 4], (2) Collaborative filtering (CF) [5–7] and (3) Hybrid filtering (HF) [8, 9]. Later, graph-based [10–12] and knowledge-based [13–15] approaches have been introduced. CBF relies on the calculation of text similarity between the paper's metadata and the user's search keywords or preferences. CF depends on the choice of a similar user group of a target user for the recommendation. On the other hand, HF combines CBF and CF to enhance the recommendation quality. The graph-based method uses a network graph to find the most relevant papers. Whereas, the knowledge-based recommendation systems are based on explicit knowledge acquisition about items, user preferences, and recommendation criteria. This approach is most suitable for the situation where sufficient knowledge about the preference of items is unavailable. Of late, a few research paper recommendation systems [8, 13, 16] have been proposed following either CBF, CF, Hybrid, Graph-based, or Knowledge-based filtering techniques. Apart from the filtering techniques, variation is also known in the design of different modules in a recommendation, such as an article representation, user profile generation, ranking candidate papers, etc. Article representation is one of the most important task for the CBF approach. In general, to represent articles, the state of the art concepts are "bag-of-words" [4, 17, 18], "vector space" [19], "key-phrase" [20] "metadata" [21, 22], "user given tag" [23], etc. To represent user profiles, existing systems mostly considered terms extracted from user's past publications or given by users [24]. Few of them used the user's preferred topic [20] for representing a user's profile. Others utilized a citation network [11] to find the common relation among users or articles. Some research works [16, 25] on paper recommendation systems deal with other metrics, namely diversity, novelty, coverage, etc. In addition, in order to rank papers, the existing works utilized distance metric [4, 11, 25, 26], similarity measure [8, 16], utility measure [3, 8], etc.

Nevertheless, the existing research paper recommendation systems are not free from issues. It may be noted that among several filtering techniques, CBF is more pertinent for the research paper recommendation system. Since a research paper is textual data, it is highly informative. In CBF based paper recommendation system, the main drawback lies in the proper representation of articles and user profiles. The existing approaches are unable to perform exhaustive analysis. Hence, some important characteristics of papers were not explored, such as the popularity of papers, the complexity of papers, papers with a new concept, scientific quality, relevancy of papers,

[1] APARAVI, Data Integration and automation, 2021: "Big Data Growth Statistics to Blow Your Mind (or, What is a Yottabyte Anyway?)": Source: https://www.aparavi.com/resources-blog/data-growth-statistics-blow-your-mind

[2] The Economic Times: "India is world's third largest producer of scientific articles: Report-2019", Source: https://economictimes.indiatimes.com/news/science/india-is-worlds-third-largest-producer-of-scientific-articles-report/articleshow/72868640.cms?from=mdr

etc. Further, the recommendation systems considered only the user's preference as the user's profile; the same was used for filtering to find candidate papers. In other words, such a system depends on capturing the user's preference (what users like) but not the user's intent (what users want). This issue is significant as user's intent may change from time to time. Though in our previous works ([27, 28]), two distinctive and effective approaches of article representation and user behavior prediction were introduced, they were not integrated into a common platform to mitigate the ultimate goal of recommendation. Further, existing paper recommendation systems did not consider any feature engineering (feature generation and feature optimization) technique to improve prediction analogy. In addition, traditional models of article recommendation mainly focused on rating prediction or simply measuring similarity scores to rank papers. There is a need for an improved ranking algorithm that can provide a better recommendation with few or no ratings. The ranking algorithm would be worthwhile if it includes the user's choice along with some utility measures to improve the result of the recommendation. In contrast, the current article recommendation systems emphasize relevancy or accuracy in the recommendation, whereas diversity, novelty, serendipity, etc. are ignored. Further, the CBS-based paper recommendation also suffers from a cold start problem which was not addressed in the existing approaches.

In view of the state-of-the-art approaches, this work aims to design a recommendation model that would fulfill the current demand of researchers and provides a better-personalized recommendation. In this direction, this work finalizes the following objectives.

- To provide a solution from a system view for designing a prototype of a research paper recommendation system.
- To utilize a proper feature engineering technique to improve the visualization of articles as well as the quality of recommendation.
- To capture dynamic user's intent for the perfect recommendation.
- To formalize a better ranking algorithm for making the recommendation personalized and improving the performance.
- To incorporate novelty and diversity along with relevancy in the recommendation.
- To overcome the cold start problem.

This paper proposes a prototype of a multi-criteria-based personalized research paper recommendation system. Hereafter, the proposed system will be referred to as SHARE. It follows content-based filtering to suggest personalized research papers to researchers. First, SHARE utilizes multiple features that find out valuable insights of papers as well as represents papers uniquely. Besides, these features are also used for representing user's profiles. Secondly, the proposed system considers what users want along with what users like to capture the dynamic notion of users. Further, to present the recommendation list, SHARE proposes a hybrid ranking strategy. The proposed ranking algorithm is a combination of MCDA (Multi-Criteria Decision Analysis) which ranks papers based on analyzing multiple features and SVMRank which is used to rank papers based on user's choices. Finally, a rank aggregation method is used to get the perfect and personalized rank list. For the first recommendation, SHARE does not depend on user's past publications or user's past research work. Hence, SHARE avoids cold start problems for new users to some extent. Also, SHARE recommends papers analyzing multiple criteria of papers. This scheme helps to discover new papers even with no citation. Thus SHARE overcome the cold start problem for new papers. Moreover, diversity and novelty are also incorporated here.

SHARE is evaluated using both system and user-based evaluation methods. In system evaluation, a comprehensive experiment is carried out using two real-life dataset, namely, "Cite-U-Like" [3] and "Scopus" [4]. For user-based evaluation, a dataset crawled from Google Scholar is considered to implement the system on large scale and increase the coverage of recommendations.

The main contribution of this paper are summarized in the following:

1. SHARE integrates a meticulous feature engineering process along with an effective dynamic user behavior prediction technique to improve the performance of the recommendation system.
2. SHARE predicts user intention considering the external context of a user for the perfect recommendation.
3. A novel ranking algorithm is proposed to retrieve personalized and most important papers.
4. SHARE incorporates diversity and novelty along with relevancy in the recommendation.
5. SHARE conducts user case studies on the online system, which allows researchers to understand this task from a system view.
6. SHARE evaluates the system on a large scale considering Google Scholar. Thus, it increases the coverage of the recommendation.

The rest of the paper is organized as follows: Section 2 reviews existing approaches to research paper recommendation system. The proposed methodology of SHARE is discussed in details in Section 3. Section 4 includes the experiments and experimental results. Section 5 presents an analysis and discussion of the work. Finally, Section 6 concludes the paper.

2. Related work

This section reviews existing research works regarding the methodology of recommendation systems specifically for textual data.

2.1. Content-based filtering (CBS)

Verma and Dhanda ([3]) presented a recommendation system for academic literature considering incremental datasets. The authors applied Probabilistic Latent Semantic Analysis (PLSA) algorithm to cluster papers based on topics. Further, papers were filtered out according to topic matching between papers and user's given topics. The authors also utilized the Efficient Incremental High-Utility Itemset Mining algorithm (EIHI) technique to find out high utility papers. The dynamic datasets used in this work helped to discover the most recent related papers. However, the topic identification used in this paper did not consider contextual relations. Hence, it may lead to improper topic classification. Lee *et al.* ([4]) proposed a personalized research paper recommendation system that recommends papers relevant to a researcher's own published paper. They also designed a web crawler to collect research papers from the web. The candidate papers in their proposed architecture are represented using the bag-of-words model. Then, the k-nearest neighbor methodology was used to find the most similar papers to the researcher's previous papers. Bulut *et al.* ([22]) presented an article recommendation system inspecting user's

[3] `https://github.com/js05212/citeulike-a`
[4] `https://www.scopus.com/search/form.uri?display=basic`

research interest. They thoroughly surveyed a researcher's interest from the past publications and searched articles to make a user profile. Finally, metadata of profiles and metadata of articles were compared to find related articles. Chandrasekaran *et al.* ([29]) proposed a document recommendation method for cite-seer users based on concept matching. In their study, the authors divided the system into three modules, namely the classifier module, profiler module, and recommender module. The classifier module represents the documents using the predefined concept vector. The profiler module creates a user profile based on the concepts collected from their published papers. The recommender module searched papers from the database for each concept presents in each user profile. Further, the concept vector of the user profile and list of documents were converted into a weighted concept tree. In their study, the authors applied the tree-edit distance measure to rank papers.

Achakulvisut *et al.* ([17]) introduced an open-source python library "Concierge" for implementing a recommendation system. The authors utilized Latent Semantic Analysis (LSA) technique to extract the topic of the papers. Further, a Rocchio Algorithm was used on the relevant and nonrelevant documents collected from user's feedback to produce query papers. The nearest neighbor algorithm was performed to obtain the ultimate suggestion.

Amara and Subramanian ([18]) suggested an efficient approach to retrieve user's information and maintaining the storage of user's data for a content-based recommendation framework using text corpus. The authors extracted tokens from the article's title, URL using NLTK and used them as a tag for articles. Further, the user profile was constructed using explicit feedback from users.

Yang ([30]) *et al.* proposed a research-related recommendation system combining BERT and LDA models. In this work, users given input sentences were embedded in a vector space model and found similar literature for recommendation by similarity calculation. This work mostly focused on providing contextually similar documents, whereas, other aspects of recommendation such as novelty, serendipity, and diversity were ignored.

Chen ([31]) *et al.* presented an empirical study on several smoothing techniques in language modeling. This work helped to reduce the data sparsity problem.

Hong ([20]) *et al.* implemented a personalized research paper recommendation system. In this work, the authors proposed a user profile-based algorithm to extract keywords using keyword inference. Although, this keyword-based representation leads to high data processing. Moreover, keywords are not enough for the content analysis of research papers.

Pazzani and Fang ([32]) discussed a content-based recommendation system. In this approach, the researcher's publications were clustered depending on topics. After which papers were recommended based on finding similar topics.

Ebesu and Fang ([33]) proposed a context-aware paper recommendation based on a citation network. The authors used a flexible encoder-decoder to represent the context of the paper. Further, they augmented a max time delay attention network and author network to recommend related papers.

Cai ([34]) *et al.* investigated a generative adversarial network to represent heterogeneous bibliographic representation. This work considered the paper's content and citation relations to learn the network representation. Finally, top papers were ranked based on similarity measures.

Xie ([35]) *et al.* introduced a paper recommendation system based on analyzing cross-domain correlation. It was a content-based approach where papers were represented as a probabilistic association with classified hierarchical domains. Moreover, the authors represented the user's interest using probabilistic distribution over the target domain.

Though, the state of the art content-based research paper recommendation systems stated different techniques to find out relevant papers based on content matching, a few drawbacks still exists. Most of the solutions considered user's publication for defining user's interests which is a constraint for new users. Few of the existing works built user profiles based on user's explicit feedback which gives an extra burden on users. Apart from that, the topic extraction techniques(LDA, LSI, PLSA) adopted in the existing solutions for representing research papers are not perfect. The aforesaid techniques are not able to find out topics based on contextual correlation.

2.2. Collaborative filtering

Wang *et al.* ([5]) suggested a collaborative topic modeling approach to recommend scientific articles. The combination of collaborative filtering with topic modeling allows a latent topic space to explain both observed topics and observed ratings. The combination of collaborative filtering with topic modeling proposed in this work efficiently differentiates articles related to different subtopics under a similar kind of topic. Thus, users from different application domains of interest were suggested with papers having different subtopics but similar topics. Nevertheless, topic extraction techniques used in this work can be improvised considering contextual topic modeling.

Alfarhood and Cheng ([6]) suggested a collaborative filtering technique integrating an attentive autoencoder to recommend scientific articles. The attentive mechanism helped to select the only relevant part of input data. Further, the compressed contextual information of articles obtained from the attentive autoencoder was passed to the probabilistic matrix factorization model to learn the latent factor of the user and article. The authors addressed the solution to two important problems of collaborative recommendation, data sparsity related to user's ratings and content analysis of a paper. Although, a user's latent factor was not explored here. Along with that, the authors did not concentrate on the ranking of papers and recommendation quality other than relevancy.

Mnih ([7]) *et al.* introduced a probabilistic matrix factorization (PMF) model for improving the performance of the collaborative filtering technique. Moreover, the model proved the better performance on large, sparse, and, imbalance data. The main drawback of this model is it is computationally expensive.

Sun *et al.* ([36]) introduced a collaborative filtering-based article recommendation model. The model focused on identifying the proper connection of researchers to find out like-minded users. Further, similar articles were extracted using the keyword matching technique.

2.3. Hybrid

Wang *et al.* ([8]) proposed a hybrid article recommendation system integrating social information from a scientific social network (SSN). The authors adopted content-based filtering to retrieve most similar papers of a researcher's interest. Moreover, collaborative filtering was used to make a similar group of researchers and identify articles that are not read by the target researcher but liked by other researchers in friends network. In the last step, two results are combined and used to improve the recommendation.

Kanakia *et al.* ([16]) designed a hybrid research paper recommendation system using Microsoft academic citation network. The authors combined content-based and co-citation-based approaches to produce related but new papers and maximize recommendation coverage, and user satisfaction. This paper concentrated on coverage, scalability, and user satisfaction.

Sun *et al.* ([21]) proposed a hybrid approach to recommend scientific articles. They explored three modules as relevancy analysis module, connectivity analysis module, and quality analysis module for designing the recommendation system. The relevancy module calculates the weight of the keywords of an article followed by constructing a correlation matrix among keywords. The keyword correlation matrix was used to match the relevancy between the user profile and the article profile. The connectivity module analysed three types of connectivity of a user such as (1) user-user, (2) user-article, and (3) user-keyword. Collaborative filtering utilizing the Random Walk with Restarting (RWR) technique was used to find the relevant article of a user. Next, the quality module judged the quality of the reference papers of a user's published paper in terms of recency, citation, and journal impact factor. Then for a target user, candidate papers are filtered out from a corpus of papers using keyword matching. Further, the candidate paper was ranked based on the weight obtained from aggregating relevancy, connectivity, and quality score.

Berkani *et al.* ([37]) proposed a hybrid article recommendation system. For representing articles and user profiles, the authors used content-based filtering. Further, a clustering-based collaborative filtering approach was utilized to improve recommendation accuracy. In this work, the authors considered the increasing volume of data for recommendation and, tried to solve the scalability issue.

Wang *et al.* ([38]) proposed another hybrid recommendation algorithm for recommending research papers to a specific research group in a Scientific Social Network (SSN). This work addressed a rule-based approach to improve group aggregation. Along with that, the authors used Probabilistic Matrix Factorisation (PMF) technique for individual prediction as PMF possessed stable performance in CF-based approaches. However, the CBF technique used in this paper needs proper content analysis, as well as user's preferences, were not analyzed thoroughly.

2.4. Graph-based

Liu *et al.*([10]) proposed a keyword-driven and popularity-aware paper recommendation system based on an undirected citation graph. An undirected paper citation graph is used to find all papers which consist of search keywords along with other papers that are highly correlated. Further, a tree was formed to find the popular paper. This approach mainly focused on finding similar papers across different domains. However, the user requirements were not analyzed properly. Also, graph-based similarity analysis increased the sparsity problem.

Xia *et al.* ([11]) presented an article recommendation method for discovering common author relationship and past preference data. Their work was divided into two tasks. The first task was to identify the researchers which tend to search articles of the same authors. The second task was to find the articles written by a common author. For selecting the target researcher, the authors performed feature extraction from the past preferred papers of that researcher. If the ratio of features value was larger than the given threshold then the researcher was considered as a target researcher. Next, to find articles with common authors, a graph was constructed based on two relations such as (1) whether a researcher shows interest or reads an article and (2) if there is a common author between articles. Further, a random walk with a restart method was employed after transition probability computation to rank articles.

Liu *et al.* ([26]) suggested also a citation-based scientific article recommendation method integrating citation relations and researcher's historical preference. It was a two-stage approach. In the first stage, the citation relation filtering module measured the degree of association between articles considering researchers' historical preferences and citation relation between articles. In this process, papers with a low association were filtered out. In the next stage, this filtered citation relation was incorporated into the graph-based paper ranking method. In the graph-based ranking, the graph was constructed using a random walk algorithm applied to the researcher-article relation obtained from the researcher's historical preference and article-article relation obtained from the first stage. Further, they calculated the transition probability of moving one node, say researcher or article to another researcher and article node. Finally, the filtered citation relation was incorporated into the graph-based paper ranking method.

Chakraborty *et al.* ([25]) designed a framework named "FeRoSA" for recommending scientific articles. For a given query paper, "FeRoSA" provides a list of relevant papers categorized in three tags such as alternative approaches, backgrounds, and methods. The authors used citations link of papers to find the relevant papers and extracted contexts such as introduction, related work, method and results, and conclusion from the citation papers. Then, the citing and cited papers were assigned to a tag depending on their context and a citation network graph was constructed. The graph was started from the node comprising query paper and expanded by discovering and linking all the citing and cited papers. The edges of the graphs were labeled with the tag. Further, a random walk with a restart algorithm was applied to find the relevant papers in one-hop and two-hop distances. Apart from the random walker, the authors used a probability factor to identify non-reachable nodes. Next, a cosine similarity was considered to calculate the similarity between the query paper and the papers in the subgraph. Finally, a rank aggregation method was applied to the papers obtained from the cosine similarity rank list and tag-wise rank list.

Ali *et al.* ([12]) proposed a graph-based paper recommendation model named PR-HNE to provide personalized recommendation. PR-HNE learned articles and researchers' information such as papers' citation, authors' collaboration, venues' information, topical relevance, and labeled information from six different network graphs. The recommendation model introduced in this paper tried to solve the cold start problem by learning semantic relationships among papers and authors to provide a personalized recommendation.

Zhu *et al.* ([39]) proposed a recommendation system for providing scientific papers. The authors emphasized unified learning of multi-sourced feature representation and the relations for citation recommendation. In parallel, the authors considered user's past writing and cited references for the recommendation.

2.5. Knowledge-based

Olaf Liadal ([13]) addressed the recommendation problem in the research paper domain. In this approach, the author used a knowledge graph to exploit semantic information among papers and researchers. Also, in this paper, a ranking algorithm was proposed to analyze venue co-publishing between authors of candidate papers and intended users.

Tarus *et al.* ([14]) introduced a hybrid knowledge-based e-learning recommendation system. The paper focused on cold start and rating sparsity problems. In this work, an ontology-based model was proposed to represent knowledge about learners and learning resources. Further, top N recommendations were generated using a collaborative filtering technique and rating similarity matching. Finally, from this top N recommendation, the final recommendation was extracted using sequential pattern mining. The author tried to generate personalized recommendations by accumulating knowledge of learners and resources with learners' historical sequential patterns.

SAMIN and AZIM ([15]) presented a case study for recommending courses and supervisors in academia. This work used the topic modeling technique to represent the course and supervisor's interests. Finally, using distance metrics, some related courses and supervisors with expertise in the same domain were found and recommended.

3. Proposed method

This section presents the detailed description of the proposed model, SHARE (**S**ystematic **H**idden **A**ttribute-based **R**ecommendation **E**ngine). SHARE provides personalized recommendations considering paper's novelty, relevancy, complexity, diversity, and user's intention at a particular time. SHARE considers a range of hidden features of research papers. These features are used to represent a paper as well as a user's profile. Consider a database, say P of size N consisting of research papers. For a given search keyword, say q provided by a specific user, u_i at time t_j, SHARE crawls the relevant papers $P_r = [{p_r}^1, {p_r}^2, \cdots, {p_r}^n]$, where, $(n < N)$, from the corpus of papers. Next, the papers are represented using m number of features, say $F = f_1, f_2, \cdots, f_m$, obtained following an efficient feature engineering process. The problem is to find out the more relevant, diversified and personalized papers $P = [{p_r}^1, {p_r}^2, \cdots, {p_r}^{n'}]$, $(n' < n)$ that matches the intention of u_i at time t_j. The ultimate goal is to rank papers using the proposed hybrid ranking strategy to present the recommendation list in a way that increases user satisfaction in terms of relevancy, novelty and personalization. Each module present in SHARE is discussed in the following subsections.

3.1. Overview of the proposed method

An overview of SHARE is shown in Fig 1. SHARE follows five basic steps.

1. Represent the database using appropriate features.
2. Generate the candidate papers concerning user's search keywords.
3. Establish a user's profile analyzing the user's clicked activity.
4. Predict the user's intention at the time of the recommendation.
5. Rank papers according to a user type(new or old).

To accomplish the above steps, the framework of SHARE is divided into different components: (1) Article representation, (2) User interface, (3) Candidate paper generation, (4) User profile representation and intention prediction, and (5) Ranking. Article representation is responsible for identifying all the appropriate features of papers and presenting them in a suitable format. The user interface is used for providing search queries, users registration, and representing recommendation results. The third module extracts relevant papers based on user-given search queries. These relevant papers are represented as a candidate paper set. The fourth component defines user's profiles with distinctive features and employes a deep learning model to capture user's demands at a particular moment before recommendation. Finally, SHARE utilizes a novel ranking strategy to rank papers for the recommendation.

3.2. Article representation

Article representation plays an important role in the content-based research paper recommendation system. The quality of recommendation depends on the exploration of information of items (i.e., papers here). In this system, article representation of paper corpus has been done offline and it is a one-time process. The necessary steps to represent articles are shown in Fig. 2 and is discussed in the following.

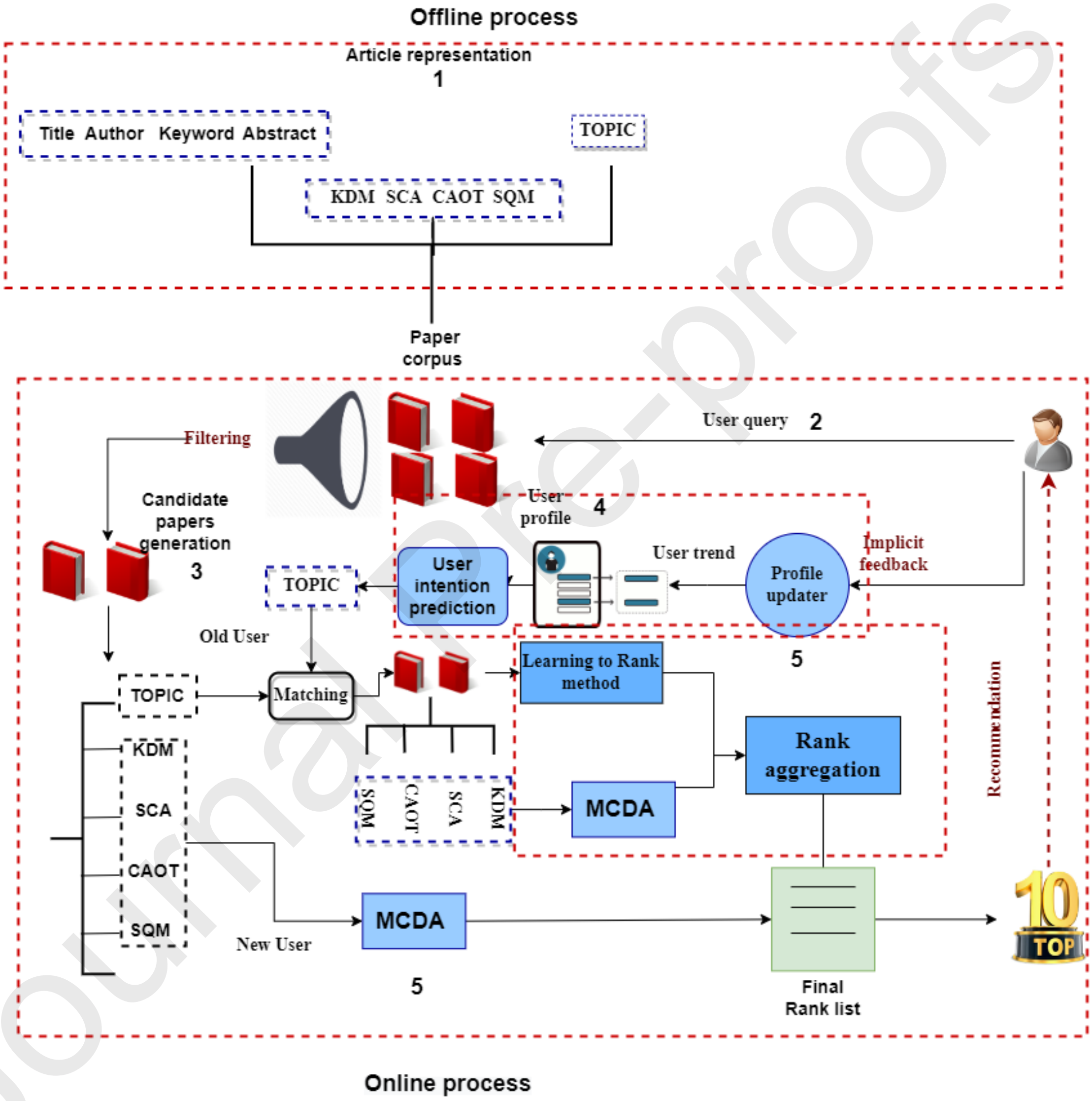


Figure (1) Framework of SHARE.

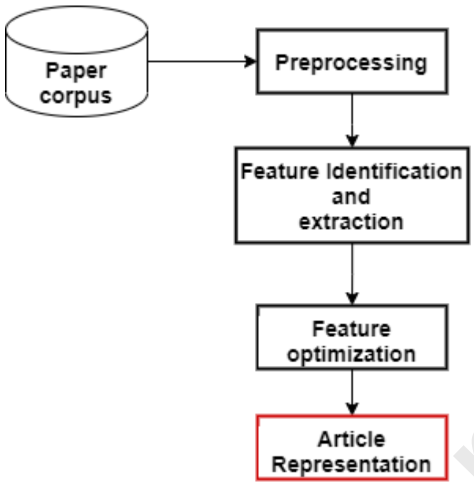


Figure (2) Overview of article representation process.

3.2.1. *Pre-processing*

In the pre-processing steps, the discrepancy of the database, such as missing values, and ambiguous information is taken into consideration. For example, the common missing value throughout the database is the keyword. To treat this missing value, keywords are generated from the abstracts of papers utilizing the Rapid Automatic Keyword Extraction (RAKE) algorithm [40].

3.2.2. *Feature identification and extraction*

After the pre-processing step, a few more features that are not present in a database, such as Keyword Diversification Measurement (KDM), Sentence Complexity Analysis (SCA), Citation Analysis Over Time (CAOT), Scientific Quality Measurement (SQM), and Topic are identified. These five features are taken into account because they can explore new insights of papers and characterize each paper in a different aspect. Moreover, the deciding features are also equally important to increase the performance of the recommendation system. The significance of the five features is mentioned as follows.

- Keyword Diversification Measurement (KDM): It scores the paper based on the degree of association among keywords. Higher association indicates lower score and vice-versa. The purpose of KDM is to find out papers with diverse topics and sub-topics. It is proposed in a recommendation to promote diversity. It will also investigate the patterns of ongoing research.
- Sentence Complexity Analysis (SCA): SCA analyses papers based on the complexity of writing style and assigns scores measuring the difficulty level.
- Citation Analysis Over Time (CAOT): CAOT calculates the score of papers analyzing the trend of popularity related to the current time.
- Scientific Quality Measurement (SQM): To determine the scientific and academic performance of the paper, SQM is used depending on the impact factor and h-index of the authors of an article in the journal.

- Topic: Topic is also an important feature for describing papers. A topic is also known as a thematic word. The concept of the whole paper can be expressed by the topic.

The extraction of feature values of KDM, SCA, CAOT, and, SQM are utilized in the same way as described in the paper [28]. The topic of each paper is decided using a hybrid topic modeling technique in a similar manner described in the paper [27]. Finally, a paper P is represented by a feature vector $F = f_1, f_2, \cdots, f_m$. Though a sufficient number of features can precisely characterize the papers, irrelevant and redundant features will create a "curse of dimensionality", over-fitting problem. It may degrade the performance of any prediction model. To conquer this problem, an Unsupervised Feature Selection (UFS) technique [41] is taken into account. This UFS method helps to find out an optimal feature set, say, $F^{'}$ with non-redundant and relevant features from the feature set F, where $F^{'} < F$. Now, The optimal feature set $F^{'}$ is used in the successive steps of recommendation.

3.3. User Interface

Users of SHARE can interact through User Interface (UI). Users have to register before accessing the system.

3.4. Candidate paper generation

Based on the user's given search keywords, in the next step, a crawler crawls a paper repository to extract papers that contain search keywords in the title, abstract, or keyword section. These retrieved papers are considered candidate papers. Further, the candidate papers are stored using their necessary features. These candidate papers are used for matching user's profiles and recommendations. After generating candidate papers, the next task depends on the type of user. If the user is new, the next step is ranking candidate papers and recommendations. For an old user, the next task is to study the user profile.

3.5. User profile representation and intention prediction

In a content-based recommendation system, implicit user preference prediction is most preferable to rating prediction. To collect user's preferences, SHARE uses user's log files that are generated from users click activity. It assumed that if a paper from the recommendation list is clicked, is preferable for that user. Afterwards, the user profile is built with the feature's values, such as "KDM", "SCA", "CAOT", "SQM" and "topic" extracted from the clicked papers. It may be noted that, "KDM", "SCA", "CAOT", "SQM" and "Topic" are common features for user profile and candidate papers. However, user preference basically indicates what users liked in the past which is static. For perfect recommendation, it is more important to capture users intention, that is, what users want at a particular time and it is dynamic. User's intentions may change in different contexts like time, location, experience, etc. To accomplish this matter, a deep learning-based user intention prediction model [27] is employed which captures the desirable topic of interest of users at a particular moment. Then, the papers with similar topic groups is used for recommendation. However, the user's intention model is sequential and made up of Long Short Term Memory (LSTM) and predicts the next two expected topic groups using past preferences of topic sequence. The detailed steps of users intention prediction are described in the Algorithm 1 and the flowchart is presented in Fig. 3.

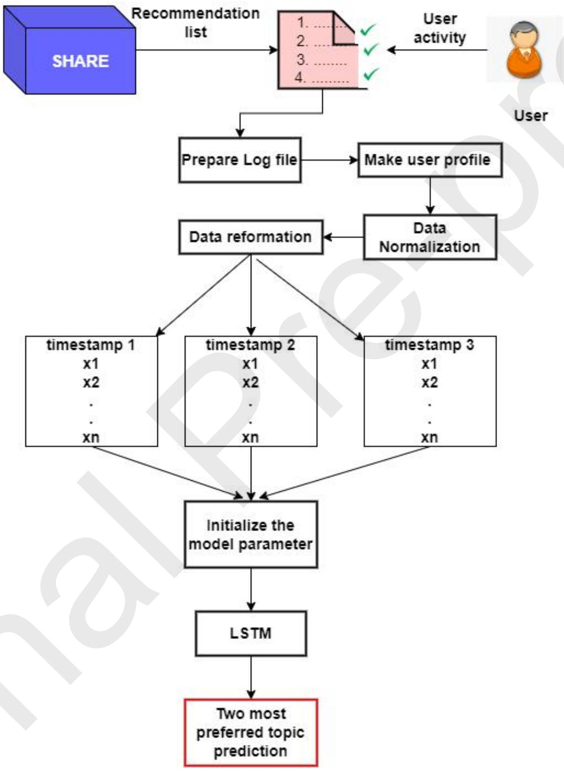


Figure (3) Flowchart of users intention prediction.

Algorithm 1: USER INTENTION PREDICTION

Input: Browsing records of M users.
Output: Two most preferred topic

```
for each user do
    Prepared user log file comprising with paper Title, Topic, KDM, SCA, CAOT, SQM,
      Date, Time between two clicked papers, Clicking order.
    Normalize the data
    Reshape the data to form batch size, timestamp, input size.
    Initialize the model parameters
    Apply LSTM to predict the next desired two topics group of the user.
end
```

3.6. Ranking and recommendation

SHARE follows a novel ranking approach that not only improves the quality of recommendation, integrates personification. SHARE applies different ranking strategies for new users and old users. For new users, papers are ranked by analyzing different criteria of papers. For old users, a new hybrid ranking method is proposed. The proposed ranking algorithm combines multiple features of paper along with user's personal choice to rank papers. Now, the process of ranking papers for recommending new as well as old users is described as follows.

3.6.1. Ranking strategy for new user recommendation:

For recommending new users, candidate papers that are retrieved based on the user-given query, are considered. These candidate papers are ranked before recommendation. To rank the candidate paper set, a Multi-Criteria Decision Analysis (MCDA) [42] technique, specifically, Simple Additive Weight (SAW) [42] is utilized. MCDA technique evaluates multiple (conflicting) criteria as a part of the decision-making process. Among the available MCDA algorithm, SAW is a simple and commonly used method. To select criteria, four features of a paper, such as, "KDM", "SCA", "CAOT", and "SQM" are considered here. These four features not only represent papers distinctively but also are equally important to analyze the quality of papers from a different aspect. Consider a database D contains a set of papers $[P_1, P_2, \cdots, P_n]$. Let, each paper P comprising of features such as $[KDM, SCA, CAOT, SQM]$. The steps to rank papers using the SAW method are described as follows.

Step 1 Design a decision matrix to present MCDA model in the following way: Where, $P_1, P_2, \cdots, P_n$

Table (1) Decision matrix for MCDA

	KDM	SCA	CAOT	SQM
P_1	X_{11}	X_{21}	X_{31}	X_{41}
P_2	X_{21}	X_{22}	X_{23}	X_{24}
....	. .	. .	. .	. .
P_m	X_{m1}	X_{m2}	X_{m3}	X_{m4}

are the feasible alternatives and $KDM, SCA, CAOT, SQM$ are the feasible criteria. X_{ij} is the value of P_i for criteria C_j, $C \in KDM, SCA, CAOT, SQM$

Step 2 Normalize the decision matrix using the following equation:

$$\frac{X_{ij}}{\max\limits_{i=1,2,\ldots,m}(X_{ij})} \tag{1}$$

Step 3 Calculate the weighted decision matrix from the normalized decision matrix. The weight of each criterion is calculated utilizing the entropy [43] method and is denoted as $H(C_j)$.

$$H(C_j) = -p\sum_{i=1}^{m} X_{ij} \cdot lnX_{ij}\ ;\quad (p = \frac{1}{ln(m)})\ and\ (1 < j < n) \tag{2}$$

Step 4 After the entropy calculation, the weight of each criteria is determined using theory of entropy [44]. According to the entropy weighted method, the larger the entropy means that the criteria provides higher information. In other words, the differentiation degree of that index is greater and the index should get a higher weight [43]. Therefore, the weight of criteria is calculated, followed by the calculation of differential degree of index.

$$DC(C_j) = (1 - H(C_j))\ (1 < j < n) \tag{3}$$

$$W(C_j) = \frac{DC(C_j)}{\sum_{j=1}^{m} DC(C_j)}\ (1 < j < n) \tag{4}$$

Where, DC stands for Differentiation Coefficient and W stands for Weight.

Step 5 Finally, the evaluation score of each alternative is calculated utilizing following equation.

$$E_i = \sum_{j=1}^{n} W(C_j)X_{ij} \tag{5}$$

After that, papers are ranked in decreasing order of evaluation score (E_i). Finally, the top 5 papers are extracted from the ranked papers and suggested to the new users. The steps to recommend new user is described in Fig. 4

3.6.2. Ranking strategy for old user recommendation:

For recommending old users, user intention in terms of the topic of interest is predicted. In this context, the first two predicted topic groups are considered. After that, few papers that belong to the same topic group as predicted user's intention are extracted from the candidate paper set. For example, if a user u_i's most preferable two topic groups at a particular moment are "2" and "3", the papers belonging to the group "2" and "3" are only considered here for further processing. In the next step, the proposed ranking algorithm is applied to this extracted papers. The proposed ranking algorithm combines two ranking strategies, namely, multi-criteria-based ranking and rankSVM [45] to increase the quality of the recommendation and personalize the recommendation. For the multi-criteria-based ranking, SAW method is applied in a similar way as discussed in the "New user recommendation" section. Further, rankSVM is applied to rank papers based on the user's preference. The rankSVM model establishes learning to rank as learning for classification on pairs of instances and addresses the classification issue using SVM.

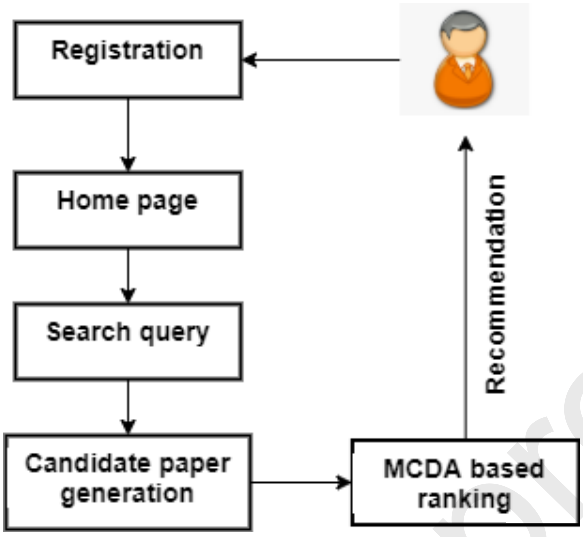


Figure (4) Flowchart of recommending new user.

Assume that there is an input space $P \in p^n$, where n is the number of documents. There exists an output space of ranks represented by $R = r_1, r_2, \cdots, r_m$, where m is the number of ranks. Let, there be the total ordering between the rank as $r_n \succ r_{n-1} \succ \cdots \succ r_1$. Here, $\succ$ denotes the preference relationship. It depends on the order of user's past click activity. Further, there exists a set of ranking functions $f \in F_r$ that determines the preference relations between two papers. Mathematically, it is expressed as:

$$\overrightarrow{f(p_i)} > \overrightarrow{f(p_j)} \; \forall \, \overrightarrow{p_i} \succ \overrightarrow{p_j} \tag{6}$$

Suppose, given a set of ranked papers $(\overrightarrow{p_i}, r_i)_{i=1}^{q}$ from the space $P \times R$. The problem is to find the best function f^* from F_r which minimizes the subjected loss function with respect to the given ranked papers. Now, consider f as a linear ranking function such that:

$$f_{\overrightarrow{w}}(\overrightarrow{p}) = \overrightarrow{w} \cdot \overrightarrow{p} \tag{7}$$

Here, $\overrightarrow{w}$ is a weight vector. It is adjusted by the learning algorithm. Now, equation 6 can be rewritten using equation 7 as follows:

$$\overrightarrow{p_i} \succ \overrightarrow{p_j} \Leftrightarrow (\overrightarrow{w} \cdot \overrightarrow{p_i} > \overrightarrow{w} \cdot \overrightarrow{p_j}) \Leftrightarrow (\overrightarrow{w} \cdot \overrightarrow{p_i} - \overrightarrow{w} \cdot \overrightarrow{p_j}) > 0 \tag{8}$$

Now, for every pair of papers, say, p_i, p_j and their corresponding ranks r_i, r_j, the label is assigned as follows:

$$\left(\overrightarrow{p_i} - \overrightarrow{p_j}, l = \begin{cases} +1 & r_i \succ r_j \\ -1 & r_j \succ r_i \end{cases}\right) \tag{9}$$

Thus from the unsupervised training dataset, a supervised training set is created. This training set is used as the classification data and an SVM model is built which assigns a label, either positive ($l = +1$) or negative ($l = -1$) to every training pairs of papers. An SVM model is constructed after introducing a slack variable ξ and solving the following quadratic optimization problem.

$$\underset{\overrightarrow{w}}{minimize}\ L(\overrightarrow{w}) = \frac{1}{2}\left\|\overrightarrow{w}\right\|^2 + C\sum_{i=1}^{l}\xi_i \tag{10}$$
$$subject\ to\ \xi_i > 0\ and\ \overrightarrow{w}.(\overrightarrow{p_i} - \overrightarrow{p_j}) \geq 1 - \xi_i\ i = 1, 2, \cdots, l$$

The above optimization problems is equivalent to the problem of classifying pairwise difference vectors $(\overrightarrow{p_i} - \overrightarrow{p_j})$ using SVM. Hence, existing SVM can be implemented to solve this problem. It may be noted that, the pair of papers are called support vectors when the constraint satisfies the equality sign that is $\overrightarrow{w}(\overrightarrow{p_i} - \overrightarrow{p_j}) = 1 - \xi_i$. When $\xi_i = 0$, it is a positively classified pair and if $\xi_i > 1$, it is a mis-ranked pair.

Further, a rank aggregation method is utilized to combine two ranking lists. To combine two rank lists, Kemeny young method [46] has been applied. According to Kemeny's young method, rank aggregation is an optimization problem that produces an ordered list of papers, say τ minimizing the sum of Kendall tau distance between two lists say, l_1 and l_2. Kendall tau distance measures the number of pairwise disagreements between two rank lists. Finally, the top 5 papers are selected and used for recommendation. The overall procedure to recommend old users is presented in Fig. 5

The flow of the recommendation process provided by SHARE is described as follows.

Input A paper corpus represented using features.

Output Top 5 recommendation.

Step 1 At first users have to register themselves providing basic information to use SHARE. After registration, a system-generated user id is provided for login and stored in the system for further processing.

Step 2 Users need to log in using a user id and password for getting a recommendation.

Step 3 Now, users can provide search keywords and ask for a recommendation.

Step 4 The crawler extracts a set of papers (candidate paper set) from the paper repository concerning the given keywords.

Step 5 In the next step, the user id is matched with the existing user id in the user profile database.

Step 6 If user id does not exist in the database, the MCDA ranking (see 3.6.1) will be applied to the candidate paper set to rank papers. From the ranked papers set top 5 papers are recommended.

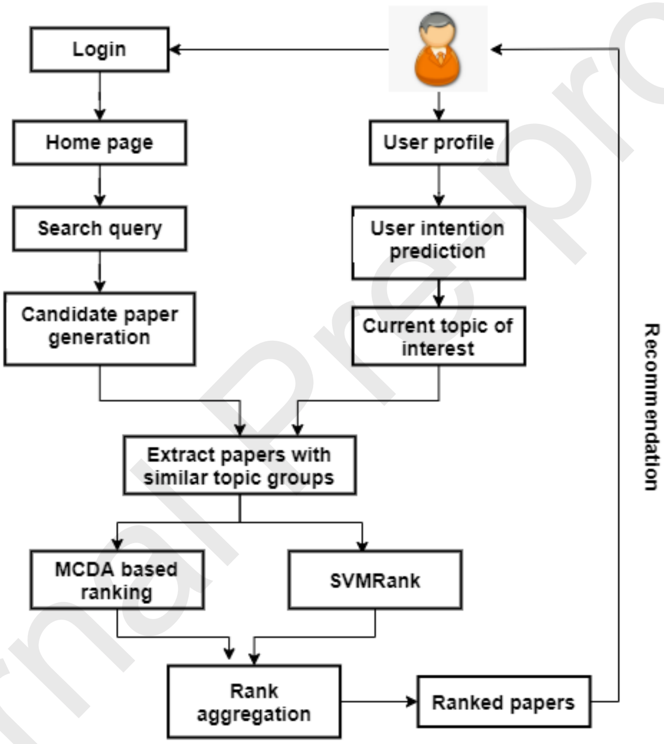


Figure (5) Flowchart of recommending old user.

Step 7 Next, from the result set, the user's click activity will be saved with the corresponding user id in a log file.

Step 8 From the log file user's profile will be created.

Step 9 On the other hand, if the user id already exists in the user database, the user intention model predicts the user's intention in terms of the topic of interest at that particular time.

Step 10 After that, a few papers are again retrieved from the candidate papers set based on topic group matching.

Step 11 In the next step, the proposed hybrid ranking algorithm will be applied to rank papers. Finally, the top 5 papers will be recommended.

Step 12 From the result set, the user's clicked activity will be analyzed again and the user profile will be updated accordingly.

4. Experiment and experimental results

This section presents the experimental objective, procedure, and experimental results to evaluate the effectiveness of the proposed recommendation system. The evaluation technique for substantiating the recommendation system is categorized into two parts: (1) system evaluation and (2) user studies.

System evaluation is a commonly used approach to test recommendation systems. Without the participation of individual users, this approach tests the accuracy of the recommendations made by a recommendation system. It uses existing information as ground truth and expects the outcomes of a recommendation system to be similar to the ground truth. The dataset is split into a training set and a test set in order to validate a recommendation method. Then, in the test set, the method tries to predict the ratings using the information from the training set. In the test sample, the expected ratings are then compared to the real ratings. Many metrics can be measured to evaluate the result.

In the evaluation based on user studies, users communicate with a recommendation framework and receive suggestions that they may or may not follow. The quality of suggestions can then be assessed by asking users for feedback or by tracking the actions of users during their interaction with the system (for example, by observing if they click on the recommended items or not). Users-based evaluations are usually time-consuming and expensive.

4.1. System evaluation

This evaluation procedure was divided into three parts. In the first part, datasets are collected and pre-processed for the evaluation. The second part mentions the metric used for the evaluation. The last part describes the experimental procedure.

Objectives: The objective of this experiment is as follows.

- To show the effectiveness of proposed SHARE with a real-life existing dataset.
- To compare the performance of SHARE with state of the art recommendation approaches.

Dataset: In this work, two public datasets (1) CiteULike [5] and (2) Scopus [6] were taken into account to evaluate the proposed SHARE. CiteULike is one of the leading Social networks for researchers to locate, store, manage, and share academic papers online with an existing social tagging [47]. The data statistics are described in Table 2.

However, the second dataset is crawled from Scopus and contains papers from computer science domain. Data statistics are described in Table 3. For this dataset, the topic was extracted from the title of papers and used as tag.

Table (2) Statistics of the dataset for experiment

Description	Value
Number of researchers	1000
Number of articles	10327
Number of tags	1050
Number of researcher-article pairs	14500

Experimental setup: All the implementation had been done in the python 3.6 environment. The web application had been developed using Flask. However, the user interface was designed using hypertext markup language (HTML) and cascading style sheet (CSS). In the process of topic extraction, LDA and word2vec were combined. The number of topics had been decided based on calculating the topic coherence score. In the word2vec model, vocabulary size was 4556, $\alpha = 0.025$, and window size was 5. Next, in the LSTM-based user intention model, the first layer was composed of 100 LSTM cells, and the second layer was composed of 50 LSTM cells. Further, the 'ADAM' optimizer was taken into account. Furthermore, in the ranking technique using RankSVM, the penalty parameter 'C' was set to 1.0.

Table (3) Data description

Properties	Description
Dataset	Scopus
Size	2000
Domain	Computer Science
Number of authors	430
Num of tags	10

Evaluation metric: For the system evaluation, precision, recall, and NDCG were used to compare the prediction accuracy of the proposed SHARE [48]. The above three metrics are commonly used in recommendation systems to assess recommendation consistency. The definition of three matrices is expressed mathematically as follows:

$$Precision@N = \frac{Number\ of\ correctly\ recommended\ papers\ in\ top\ N}{Number\ of\ recommended\ papers} \tag{11}$$

$$Recall@N = \frac{Number\ of\ correctly\ recommended\ papers\ in\ top\ N}{Number\ of\ collected\ papers} \tag{12}$$

[5] https://github.com/js05212/citeulike-a
[6] https://www.scopus.com/search/form.uri?display=basic

where "Number of correctly recommended papers" refers to the number of papers in both the Recommendation List derived from the Recommendation Method and the test results. "Number of recommended papers" is the total number of recommended papers while "Number of extracted papers" is the total number of papers of testing data. In this study, precision was used to evaluate the proportion of users' interests in the research papers recommended by SHARE. On the other side, recall calculates the ability of SHARE to recommend the right articles to the researchers, that is, it examines the degree of satisfaction of SHARE users.

$$NDCG@N = \frac{DCG@N}{IDCG@N},\ DCG@N = \sum_{i=1}^{N} \frac{rev_i}{log_2(i+1)},\ IDCG@N = \sum_{i=1}^{N} \frac{1}{log_2(i+1)}$$

where, '$rev_i' = 5$, if $i-th$ paper was clicked by users, otherwise,$'rev_i' = 0$.

Experimental procedure: In this experiment, raw data from the collected dataset was used for representing articles. Further, user data of the collected dataset was divided into a training dataset and a testing dataset randomly in an 80:20 ratio. Since each column in a matrix corresponds to articles, a set of articles in the test matrix were disjointed from those in the training matrix. Information in the training matrix was used to create the user's profiles along with content and social information. Then SHARE was used to make predictions for each user. The experiment was performed 10 times, each time the dataset was divided using different random splits.

Now, for the performance comparison of the proposed SHARE on two datasets, a total of twelve article recommendation methods were considered. Among them, the first eight were considered for comparison on the CiteULike dataset, and the last four were taken into account for comparison on the Scopus dataset. For the CiteULike dataset, comparing approaches are listed as follows:

1. Language Modeling (LM) [31]: In this approach, queries were converted into words, then the sequence of words was formed. Now, the data sparsity problem was solved using the smoothing data method.
2. Personalized Research Paper Recommendation System (PRPRS) [20]: PRPRS has been proposed by Hong et al. The authors implemented a user profile-based approach. The user profile was designed using keywords extracted from the keyword extraction and keyword inference module.
3. Probabilistic Matrix Factorization (PMF) [7]: PMF employed the matrix factorization technique on the user-item rating matrix and again utilized them to make a better prediction.
4. Social Aggregation Recommendation (SAR) [36]: SAR was proposed by Sun et al. The Authors used three types of social connections of researchers to find similar kinds of people. The authors analyzed the semantic content of articles and user's social relations to recommend them.
5. Common Author relation-based Recommendation (CARE) [11] : CARE recommended articles finding common relations among authors. CARE was also reviewed in the section 2.
6. HAR-SI [8]: HAR-SI is a hybrid recommendation system proposed by Wang et al.. HAR-SI utilized scientific social networks to collect necessary information from researchers and articles. HAR-SI is discussed in section 2.
7. CBF_{LM-NLP} [37]This approach proposed a hybrid paper recommendation system improvising CBF and CF algorithm.

8. $GPRAH_{ER}$ [38]This approach proposed a hybrid paper recommendation system to suggest appropriate research papers to the intended group of researchers.

Among eight approaches, the first two are based on content-based filtering. The next approach follows CF filtering and the last five are hybrid techniques. The primary reason for considering the above methods was that all the methods were proposed for recommending research articles. Secondly, the methods also followed similar types of objectives as the proposed approach, such as personalization, representation of paper's content, improving relevancy, novelty, and so on. The results of the first four approaches were taken from the paper HAR-SI[8], as the proposed approach considered the same dataset and metrics as described in HAR-SI. However, the results of other methods are taken from the original papers.

Further, to compare the performance of SHARE on Scopus Dataset, four states of the approaches were considered. They are listed as follows.

1. $TopicRec$ [32] TopicRec is content-based research paper recommendation system proposed by Pazzani. TopicRec is also described in sec 2.
2. $GANRec$ [34] GanRec is a generative adversarial network to represent heterogeneous bibliographic representation described in sec 2.
3. $NCNRec$ [33] NCNRec is a context-aware paper recommendation based on a citation network.
4. CPR [35] CPR was a research paper recommendation utilizing cross-domain correlation.

These approaches followed content-based filtering like SHARE and also focused on content learning and user profile learning.
To prove the significance of the result produced by SHARE compared to the results obtained from the existing state of the art methods, a paired t-test was performed considering a 95% confidence interval. The result of the statistical test is represented using the p-value. The bold p-value indicates statistically significant better behavior.

Experimental results: In this section, all the experimental results obtained from SHARE

Table (4) Performance comparison of proposed SHARE and other state of the art approaches

Approach	Precision					Recall				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
LM	0.0793	0.0612	0.0519	0.0433	0.0421	0.0752	0.1124	0.1373	0.1579	0.1752
PRPRS	0.0801	0.0729	0.0683	0.0599	0.0571	0.1032	0.1263	0.1458	0.1628	0.1783
PMF	0.0543	0.0418	0.0363	0.0341	0.0302	0.0595	0.0729	0.0873	0.0991	0.1124
SAR	0.1355	0.0997	0.0915	0.0811	0.0753	0.1334	0.1825	0.2109	0.2476	0.2668
CARE	0.1416	0.1148	0.0987	0.0892	0.0814	0.1486	0.1994	0.2335	0.2650	0.2911
HAR-SI	0.1593	0.1271	0.1105	0.0993	0.0910	0.1617	0.2185	0.2591	0.2829	0.3110
CBF_{LM-NLP}	0.052	0.0746	▽	▽	▽	0.0854	0.0748	▽	▽	▽
Improved CBF_{LM-NLP}	0.0548	0.0806	▽	▽	▽	0.1645	0.0971	▽	▽	▽
GPRAH_ER	**5.001**	**3.619**	**2.953**	**2.538**	**2.250**	1.267	1.824	2.275	2.593	2.917
SHARE	0.1882	0.1352	0.1262	0.1013	0.1002	**0.1801**	**0.2218**	**0.2711**	**0.2893**	**0.3924**

Note: ▽ implies that the approach did not report the metric.

using the dataset CiteULike and Scopus are presented and compared with the other methods mentioned in the previous section. Table 4 presents the performance of SHARE and other methods in terms of precision@N (N= 10, 20, 30, 40, 50) and Recall@N(N= 10, 20, 30, 40, 50) on CiteULike dataset. The best results are marked in bold.

From Table 4, it has been observed that with a higher number of recommendations (N), precision results are declined but the recall results are increased for all methods. However, in all cases (N= 10, 20, 30, 40, 50), SHARE achieved the highest performance in comparison to most of the methods. In particular, it may be noted that SHARE produced the best result at a lower number of recommendations. The improved results have been obtained due to the combination of proper article representation with an effective user modeling technique. Further, the proposed ranking technique is essential to increase the impact of recommendations.

Moreover, from Table5, it is observed that proposed SHARE significantly better than other

Table (5) p-value obtained on comparison of precision and recall using pair t-test at α=0.05

Metrics	Proposed model	LM	PRPRS	PMF	SAR	CARE	HAR-SI	CBF_{LM-NLP}	Improved CBF_{LM-NLP}	$GPRAH_{ER}$
Precision	SHARE	0.001	0.006	0.001	0.003	0.013	0.04	$\triangle$	$\triangle$	0.002
Recall	SHARE	0.002	0.005	0.002	0.017	0.04	0.16	$\triangle$	$\triangle$	0.001

approaches.

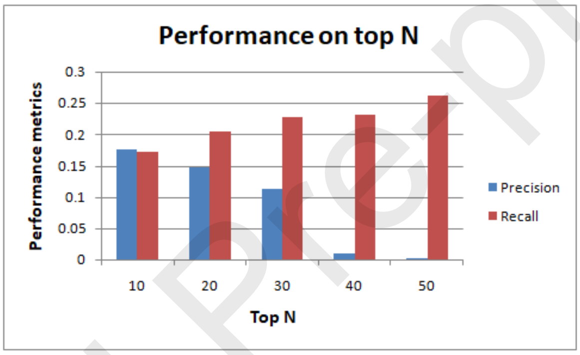


Figure (6) Performance on Scopus dataset.

Further, Figure 6 represents the NDCG results of SHARE on Scopus dataset. Table 6 presents the comparison results in terms of NDCG@N(N=20,30,40) on Scopus dataset. The results of existing methods are taken from the paper [35]. From the results, it has been observed that though SHARE did not produced best result on Scopus dataset but is better than most of the approaches.

Table (6) Performance comparison of Scopus dataset

Approaches	ndcg		
	@20	@30	@40
TopicRec	0.706	0.68	0.65
GANRec	0.754	0.721	0.71
NCNRec	0.76	0.747	0.713
CPR	0.793	0.786	0.773
SHARE	0.779	0.752	0.729

Table (7) p-value obtained on comparison of NDCG using paired t-test at α=0.05

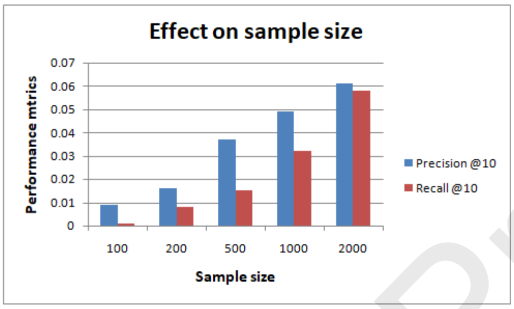
Proposed model	TopicRec	GANRec	NCNRec	CPR
SHARE	0.0008	0.01	0.05	0.07

4.1.1. Sensitivity analysis on hyper parameters:

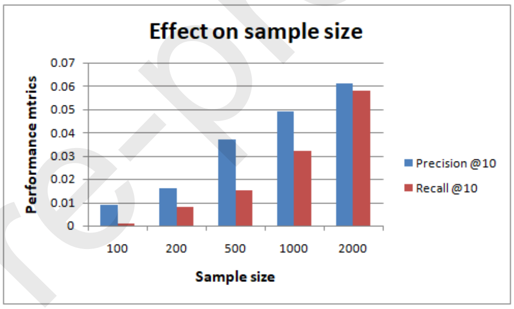
This section examines the influence of different key parameters used in this paper. In this context, impact of (1) Samples and dimensions, and (2) hyper parameter used in topic modeling, user intention module are analyzed.

Effect of samples: The results on two datasets are presented in Fig. 7a and 7b presents the performance of SHARE on different sample size. It is observed that the metric score increases with the increasing sample sizes.

Number of components for LDA: In the topic identification using LDA, coherence score



(a) CiteUlike data.



(b) Scopus data

Figure (7) Model performance on sample size.

was used to decide optimal number of topics. The model was run multiple times. Finally, the highest coherence score was obtained at number 5.

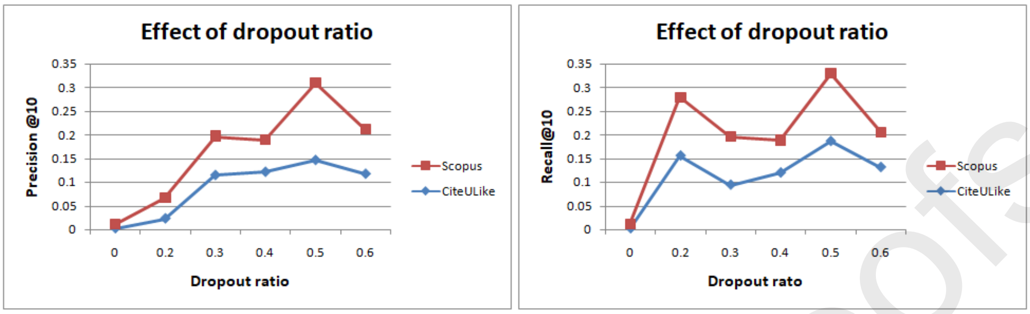
Parameters for Word2Vec: The parameter for Word2Vec is also important. After multiple epochs, window size=5 and α = 0.025 were set for best performance.

Effect of hidden layer: The user intention prediction module based on LSTM was tested with different number of nodes in hidden layers. The best performance achieved with 100 nodes in hidden layers.

Effect of dropout layer in LSTM: Further, the module was investigated with drop out rate 0.2, 0.3,0.4 and 0.5. Finally, 0.5 was set as the optimal number as it was very effective. Figure 8a and Figure 8a represents the results based on drop out ratio.

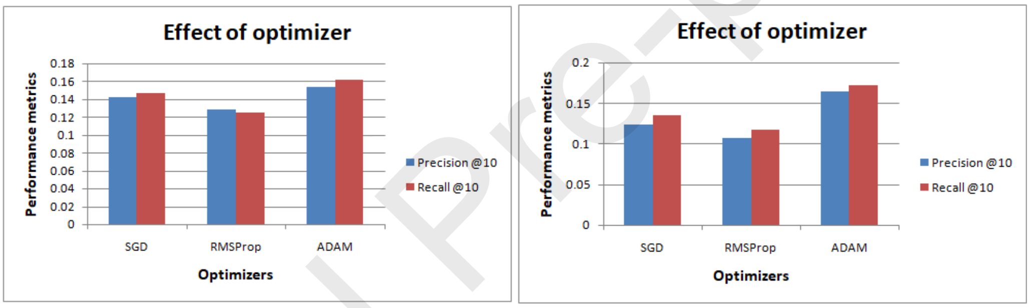
Effect of optimizer: In the decision of perfect optimizer setting, SGD(stochastic Gradient Descent), RMSprop and ADAM were taken into account. From the Figure 9a and 9b, it has been observed that ADAM was more effective than others for the underlying problem.

Effect of activation function: Figure 10 compares the performance of the system with different activation functions, namely, sigmoid, relu and tanh. But model performed better using sigmoid. Figure 10a and Figure 10b also prove the statement.



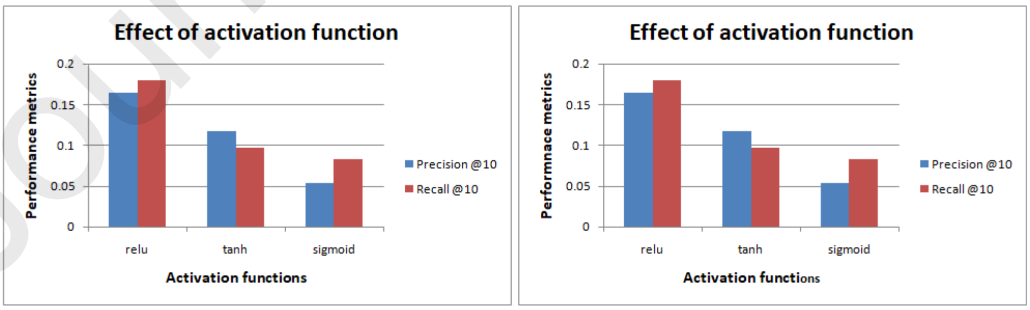
(a) Precision analysis based on dropout ratio. (b) Recall analysis based on dropout ratio

Figure (8) Model performance analysis.



(a) Effect of optimizer on citulike dataset. (b) Effect of optimizer on scopus dataset.

Figure (9) Model performance analysis.



(a) Effect of activation on citulike dataset. (b) Effect of activation on scopus dataset.

Figure (10) Model performance analysis.

4.1.2. Ablation study

In this section, an ablation study was conducted to see the interactions of the components in SHARE and validate the contribution of each component to the final recommendation. In this regard, four variants of SHARE, (1) SHARE-W, (2) SHARE-F, (2) SHARE-U, and (3) SHARE-R were considered.

SHARE-W: SHARE-W is like a search engine. It comprises of direct features(which are available in the paper itself) only. SHARE-W provides result based on similarity matching between candidate papers and search query. Top 5 from the results were provided as recommendation.

SHARE-F: In SHARE-F, indirect features(generated from direct features) were added.

SHARE-U: SHARE-U added user intention module with SHARE-F.

SHARE-R: Finally, the proposed ranking strategy in SHARE-R were investigated to find the impact of an efficient ranking in recommendation and improvement in personalization.
However, variants of SHARE are compared on the basis of precision@N(N=10,20,30,40) and recall@N(N=10,20,30,40). In the previous experimental results (Table 4 and Table 6), it has been observed that SHARE performs better in lower values of N. Hence, the max value of N=40 is considered here.

Experimental results: The comparison results of variants SHARE model on CiteULike and

	Precision	SHARE-W	SHARE-F	SHARE-U	SHARE-R
CiteULike	@10	0.043	0.074	0.152	0.188
	@20	0.027	0.053	0.121	0.135
	@30	0.018	0.048	0.103	0.126
	@40	0.009	0.039	0.092	0.101
SCOPUS	@10	0.038	0.058	0.141	0.175
	@20	0.027	0.049	0.128	0.147
	@30	0.011	0.032	0.101	0.113
	@40	0.002	0.003	0.007	0.009

Table (8) Comparison results on precision of variants SHARE model in ablation study.

	Recall	SHARE-W	SHARE-F	SHARE-U	SHARE-R
CiteULike	@10	0.104	0.125	0.152	0.180
	@20	0.122	0.153	0.186	0.221
	@30	0.144	0.192	0.248	0.271
	@40	0.152	0.189	0.267	0.289
SCOPUS	@10	0.088	0.134	0.161	0.172
	@20	0.118	0.142	0.190	0.205
	@30	0.132	0.177	0.212	0.227
	@40	0.141	0.188	0.220	0.231

Table (9) Comparison results on recall of variants SHARE model in ablation study.

Scopus dataset are presented in Table 8 and Table 9. From the results, it has been observed that the addition of successive components improves the performance of SHARE significantly. However, among all variants, user intention model performs best. Moreover, the considered features and ranking mechanism also proves the effectiveness on recommendation.

4.2. User studies

An experiment was conducted online to evaluate SHARE based on the user's opinion. For this experiment, a prototype of SHARE was developed. However, as it is a desktop application, it is not connected to a server. The implementation details is available in Github [7]. The home page of SHARE is shown in Fig. 11.

Objectives: The objectives of the user studies have been decided as follows.

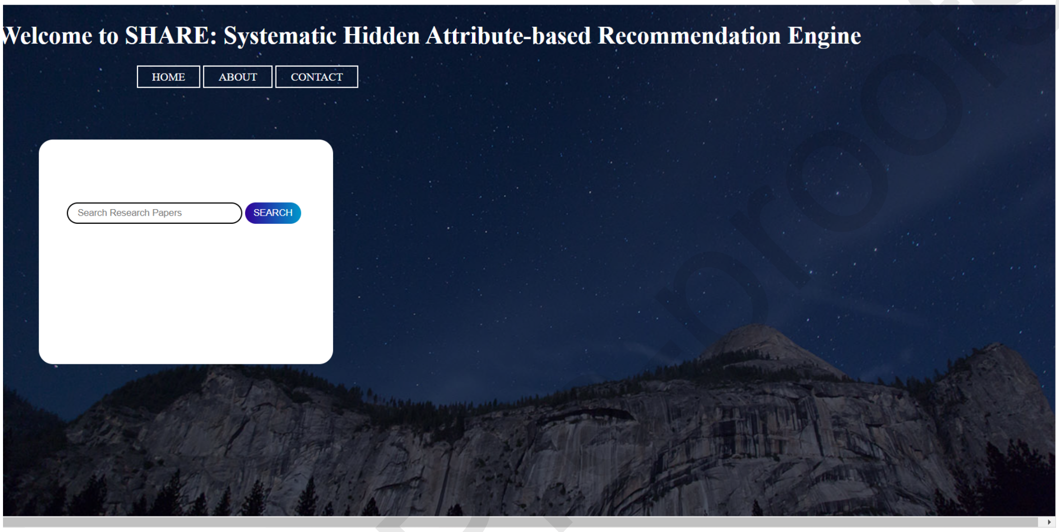


Figure (11) Home page of SHARE.

- To show the significance of SHARE based on user's opinions.
- To prove that SHARE provide significantly better recommendation than existing search engines provides in general.

Dataset: For user-based evaluation, Google Scholar was considered as a data source. Google Scholar was considered here to increase the search coverage and enlarge the variation of the domain. A web crawler was designed to retrieve the admissible papers from Google Scholar concerning the search query. The crawler considers the papers in PDF format only. Further, the crawler extracted the required data such as 'title', 'abstract', 'keywords' (if it exists) for each paper from the publisher's site. In this task, it is not required to parse the full text of a PDF. The above-mentioned data were almost available on the publisher's page of the corresponding paper. Now, the extracted data was stored in a datasheet and pre-processed. In the next step, all required features were extracted and added to the result. However, for this experiment, no local database was maintained. Feature values were extracted for each search result that was obtained against the search query. As result, it may increase the response time.

Participants: A online user study was conducted through institute mail. For this experiment, participants were selected from the Indian Institute of Technology, Kharagpur, and the Indian Institute of Technology, Kanpur. Participants were rewarded with 100 Rs. They belong to different

[7] `https://github.com/A-chaudhuri/SHARE`

domains of subject and different levels of experience. At first, a link to a google form was distributed using the connection of the authors' mail to collect willing participants. The Google form contains fields such as Name, Institute name, Research Interest, and Experience. Since the participants need to make a judgment regarding the recommendation. Participants were selected intuitively considering their familiarity with papers. Finally, 112 participants were selected for this experiment.

Experimental procedure: A online procedure had been carried out to instruct participants, evaluate the systems, and to take feedback. After the selection process, participants were informed about their slots through the mail. For each participant, 13 slots were given. In each slot, they were asked to provide twenty search keywords. Only in the first slot, before giving a query, do they need to register themselves by clicking the signup option on the home page of the SHARE. After the successful full registration, they were provided with a system-generated User ID. Further, they needed to log in by inputting their User ID and Password (chosen by the participants during registration). Now, participants could put a query on the search field. At the end of each session, the user log file was used by the system to collect the necessary information about the participant and was processed for creating a profile. Afterward, all the required features were extracted from the participant's clicked papers according to the description in subsection 3.5. Moreover, at the end of each session, participants were asked to fill out a feedback form where they were to give answers to a few questions against each recommendation page. This feedback was only collected for the evaluation of SHARE. Table 10 lists the questions for analysing users opinion. The participants had to answer it in terms of the 1 – 5 Likert scale. 1 means least score

Table (10) Assessment of the system based on user perception

Subjective variables	**Questions(each answered on a 5-point Likert scale from 1 "least score" to 5 "best score"**
Relevancy	*The recommendation is how much relevant to you?*
Novelty	*How many numbers of papers among recommended papers are unknown to you?*
Quality	*How much score you will like to give regarding the quality of recommendation?*
Flexibility	*How much score you would like to give about the flexibility of the system?*
Response time	*How quickly the system respond to you?*

and 5 means the highest score. Finally, the average marks of 13 sessions were calculated and treated as a final score of a user.

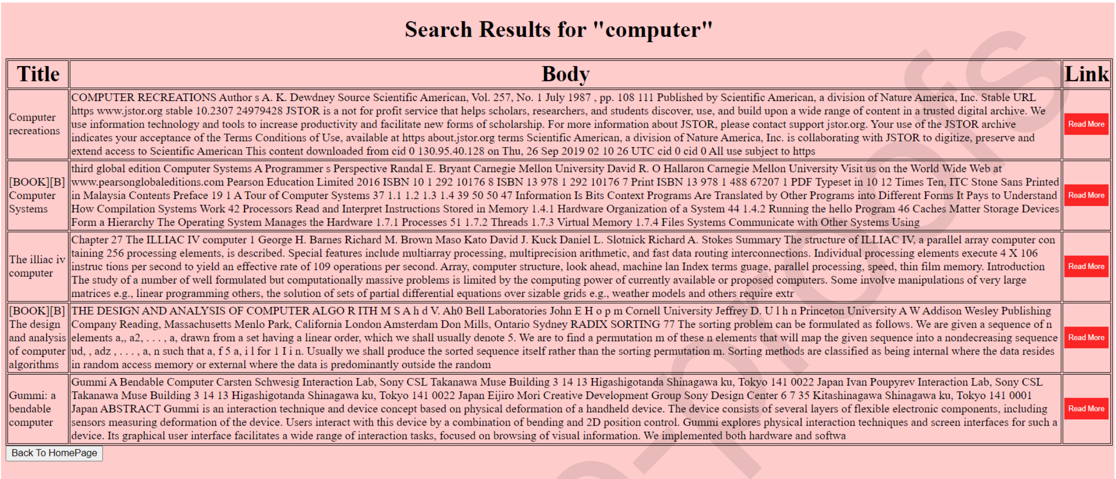
Result: Fig. 12 presents top 5 results of a query generated from SHARE.

This result was evaluated through users opinion. Fig. 13 presents the distribution of users score on different metric according to Likert scale.

From Fig. 13, it has been observed that, though SHARE for new users did not get a good response according to relevancy and quality of recommendation, SHARE for old users was considered quite good. Additionally, SHARE has proved that the recommended papers are novel for both users. After all, SHARE is flexible with a moderate response time.

Further, to check the efficiency of SHARE, it was compared with three search engines, namely, (1) Google Scholar(GS), (2) Microsoft Academic Search (MAS), and (3) Citeseer. These are currently the most popular search engines among readers for seeking research papers. Following metrics are considered to evaluate the results.

1. **Relevancy** = It is obtained from user's given scores in feedback form.

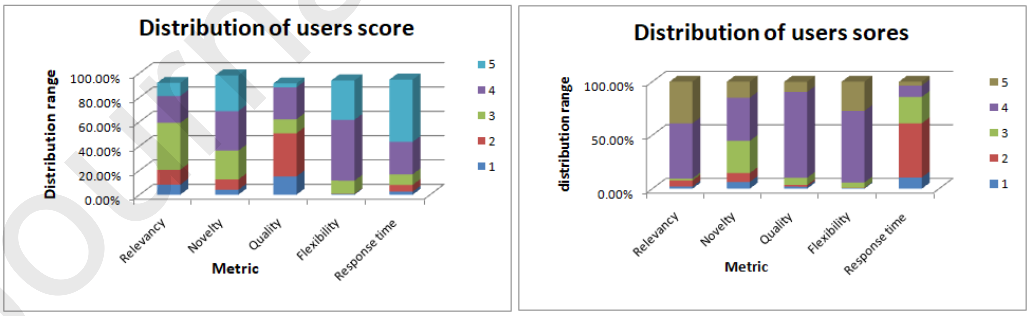


Search Results for "computer"

Title	Body	Link
Computer recreations	COMPUTER RECREATIONS Author s A. K. Dewdney Source Scientific American, Vol. 257, No. 1 July 1987 , pp. 108 111 Published by Scientific American, a division of Nature America, Inc. Stable URL https www.jstor.org stable 10.2307 24979428 JSTOR is a not for profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support jstor.org. Your use of the JSTOR archive indicates your acceptance of the Terms Conditions of Use, available at https about.jstor.org terms Scientific American, a division of Nature America, Inc. is collaborating with JSTOR to digitize, preserve and extend access to Scientific American This content downloaded from cid 0 130.95.40.128 on Thu, 26 Sep 2019 02 10 26 UTC cid 0 cid 0 All use subject to https	Read More
[BOOK][B] Computer Systems	third global edition Computer Systems A Programmer s Perspective Randal E. Bryant Carnegie Mellon University David R. O Hallaron Carnegie Mellon University Visit us on the World Wide Web at www.pearsonglobaleditions.com Pearson Education Limited 2016 ISBN 10 1 292 10176 8 ISBN 13 978 1 292 10176 7 Print ISBN 13 978 1 488 67207 1 PDF Typeset in 10 12 Times Ten, ITC Stone Sans Printed in Malaysia Contents Preface 19 1 A Tour of Computer Systems 37 1.1 1.2 1.3 1.4 39 50 50 47 Information Is Bits Context Programs Are Translated by Other Programs into Different Forms It Pays to Understand How Compilation Systems Work 42 Processors Read and Interpret Instructions Stored in Memory 1.4.1 Hardware Organization of a System 44 1.4.2 Running the hello Program 46 Caches Matter Storage Devices Form a Hierarchy The Operating System Manages the Hardware 1.7.1 Processes 51 1.7.2 Threads 1.7.3 Virtual Memory 1.7.4 Files Systems Communicate with Other Systems Using	Read More
The illiac iv computer	Chapter 27 The ILLIAC IV computer 1 George H. Barnes Richard M. Brown Maso Kato David J. Kuck Daniel L. Slotnick Richard A. Stokes Summary The structure of ILLIAC IV, a parallel array computer con taining 256 processing elements, is described. Special features include multiarray processing, multiprecision arithmetic, and fast data routing interconnections. Individual processing elements execute 4 X 106 instruc tions per second to yield an effective rate of 109 operations per second. Array, computer structure, look ahead, machine lan Index terms guage, parallel processing, speed, thin film memory. Introduction The study of a number of well formulated but computationally massive problems is limited by the computing power of currently available or proposed computers. Some involve manipulations of very large matrices e.g., linear programming others, the solution of sets of partial differential equations over sizable grids e.g., weather models and others require extr	Read More
[BOOK][B] The design and analysis of computer algorithms	THE DESIGN AND ANALYSIS OF COMPUTER ALGO R ITH M S A h d V. Ah0 Bell Laboratories John E H o p m Cornell University Jeffrey D. U l h n Princeton University A W Addison Wesley Publishing Company Reading, Massachusetts Menlo Park, California London Amsterdam Don Mills, Ontario Sydney RADIX SORTING 77 The sorting problem can be formulated as follows. We are given a sequence of n elements a,, a2, . . . , a, drawn from a set having a linear order, which we shall usually denote 5. We are to find a permutation m of these n elements that will map the given sequence into a nondecreasing sequence ud, , adz , . . . , a, n such that a, f 5 a, i l for 1 I i n. Usually we shall produce the sorted sequence itself rather than the sorting permutation m. Sorting methods are classified as being internal where the data resides in random access memory or external where the data is predominantly outside the random	Read More
Gummi: a bendable computer	Gummi A Bendable Computer Carsten Schwesig Interaction Lab, Sony CSL Takanawa Muse Building 3 14 13 Higashigotanda Shinagawa ku, Tokyo 141 0022 Japan Ivan Poupyrev Interaction Lab, Sony CSL Takanawa Muse Building 3 14 13 Higashigotanda Shinagawa ku, Tokyo 141 0022 Japan Eijiro Mori Creative Development Group Sony Design Center 6 7 35 Kitashinagawa Shinagawa ku, Tokyo 141 0001 Japan ABSTRACT Gummi is an interaction technique and device concept based on physical deformation of a handheld device. The device consists of several layers of flexible electronic components, including sensors measuring deformation of the device. Users interact with this device by a combination of bending and 2D position control. Gummi explores physical interaction techniques and screen interfaces for such a device. Its graphical user interface facilitates a wide range of interaction tasks, focused on browsing of visual information. We implemented both hardware and softwa	Read More

Back To HomePage

Figure (12) Result of SHARE recommendation.



(a) NEW users (b) OLD users

Figure (13) Distribution of users rating in Likert scale based on users feedback.

2. **Precision :** Precision refers to the number of papers liked by the users among recommended papers. That is,

$$Precision = \frac{Number\ of\ clicked\ papers\ in\ top\ N}{Number\ of\ recommended\ papers}. \tag{13}$$

3. **Novelty:** Novelty is defined by how many papers are unknown to users. that means

$$Novelty = \frac{Number\ of\ unknown\ papers}{Number\ of\ recommended\ papers} \tag{14}$$

4. **Diversity:** Diversity refers to the dissimilarity between the pair of recommended papers. Dissimilarity is calculated using the distance measure. In this experiment, cosine similarity is taken into account as a distance measure. Finally, maximum dissimilarity is considered as dissimilarity of the recommended list. Therefore,

$$Diversity = 1 - cosine(p_i, p_j)\ where, i \neq j \tag{15}$$

5. **Quality:** Quality of recommendation was judged by users and the results were obtained from user's given scores in a feedback form.
6. **CTR:** It measures how many recommended papers were clicked by the user.
7. **Response time (RT):** RT is generated by the system. It indicates the time it takes for a decision to be made between the time a query is sent and the time it receives its first answer.

The results of search engines were also collected following the same procedure as SHARE.

Figure 14 represents the comparison results obtained from SHARE and the three search

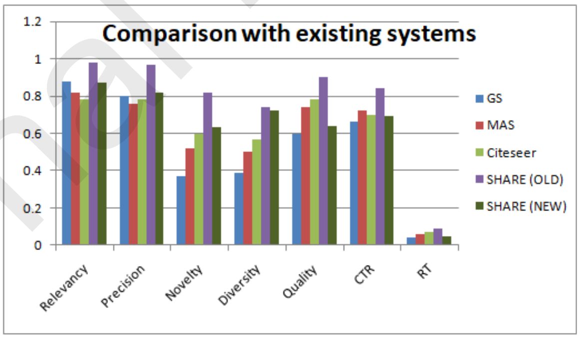


Figure (14) Summary of comparison existing systems.

engines. The comparison result proves that the proposed SHARE (both new and old users) outperforms all the state-of-the-art search engines.

Next, to prove the results obtained from SHARE, after user evaluation, is significantly different from the search engine-based results, one-way ANOVA ([49]) and Tukey's pairwise comparison ([50]) were utilized. The significance test was applied to the average result of three metrics, namely, relevancy, novelty, and quality obtained from users given feedback for all systems. From the one-way ANOVA result (Fig. 15), it has been statistically proved that there was a significant difference among the means of the four considered systems (GS, MAS, Citeseer, and

Means

Factor	N	Mean	StDev	95% CI
GS	30	8.467	1.548	(7.857, 9.077)
MAS	30	7.533	1.814	(6.923, 8.143)
Citeseer	30	7.067	1.964	(6.457, 7.677)
SHARE (OLD)	30	9.800	1.789	(9.190, 10.410)
SHARE (NEW)	30	8.333	1.241	(7.723, 8.943)

Pooled StDev = 1.69027

Figure (15) Summary of one-way ANOVA.

Tukey Pairwise Comparisons

Grouping Information Using the Tukey Method and 95% Confidence

Factor	N	Mean	Grouping	
SHARE (OLD)	30	9.800	A	
GS	30	8.467	B	
SHARE (NEW)	30	8.333	B	
MAS	30	7.533	B	C
Citeseer	30	7.067		C

Means that do not share a letter are significantly different.

Figure (16) Summary of Tukey post-hoc test.

SHARE(OLD)) at $\alpha = 5\%$ (significance level).

Afterward, Tukey pairwise comparison analysis (Fig. 16) revealed that SHARE (OLD) was highly significant than the state-of-the-art search engines.

5. Analysis and discussion

The proposed implementation strategy is compared to the 18 state of the art approaches. Table 11 summarizes the important insight gained by the proposed approach and compared it with existing approaches. From the summary, it has been observed that the proposed SHARE properly justifies the content analysis problem of CBF approaches. Further, it successfully captures user's demands promptly and provides personalized papers. Moreover, it solves the cold start problem along with incorporating various aspects of recommendation, namely, diversity, novelty, relevancy. However, user studies and system evaluation, both experiments prove the efficiency of the proposed system as well as user's satisfaction towards the fulfillment of their needs.

Table (11) Comparative analysis of state of the research paper recommendation system

Literature	Methodology	Dataset repository	Data cleaning	Feature extraction	New feature generation	Feature selection	New user recommendation	Updating profile	Ranking method	Diversity	Serendipity
[8]	Hybrid	CiteULike	Yes	Vector of words	No	No	No	No	Weight combination	No	No
[3]	Content-based	ACL anthology	Yes	References, publisher, publishing date	No	No	No	No	Utility measure	No	No
[4]	Content-based	ACM, IEEE	Yes	Bag-of-words	No	No	No	No	K-nearest neighbor	No	No
[16]	Hybrid	Microsoft Academic Graph	No	Vector of words	No	No	Yes	No	Sorting of co-citation score and similarity measure	No	Yes
[21]	Hybrid	Scholarmate	No	Keyword, publication year, citation, impact factor	No	No	No	No	keyword correlation matrix	No	No
[5]	Collaborative	CiteULike, Mendely	Yes	Topic of papers	Yes	No	No	No	*	No	No
[10]	Graph-based	Hep-Th	No	keywords	No	No	Yes	No	*	No	No
[11]	Graph-based	CiteULike	No	Author	Yes	No	No	No	Random walk with restart	No	No
[26]	Graph-based	CiteULike	Yes	Citation information	No	No	No	No	Random walk with restart	No	No
[22]	Content-based	ACM, IEEE, DBLP	Yes	Title, keyword, abstract, author, year	No	No	No	No	Sorting cosine similarity, TF-IDF	No	No
[6]	Collaborative	CiteULike	Yes	Title, abstract	No	yes	No	No	Sorting prediction score	No	No
[25]	Graph-based	ACL anthology	Yes	Citation information	No	No	No	No	Random walk with restart and sorting of similarity measure	Yes	No
[29]	Content-based	Citeseer	No	Concept	No	No	No	Yes	tree-edit distance	No	No
[17]	Content-based	Neuroscience 2015 conference dataset	No	Bag-of-words	No	No	No	No	Nearest-neighbor	No	No
[18]	Content-based	△	No	Bag-of-words	Yes	No	No	No	No	No	No
[33]	Graph-based	RefSeer	No	citation	No	No	No	No	No	No	No
Proposed approach	Content-based	CiteULike	Yes	Title, author, keywords, abstract, KDM, SCA, CAOT, SQM,Topic	Yes	Yes	Yes	Yes	MCDA,SVMrank	No	Yes

Threats to validity: All the reported experiments and results established in this paper are subject to some threats and assumptions, they are mentioned below.

- Results may change with different environments and different datasets used.

- The response time of SHARE is quite delayed while considering Google Scholar. There are two reasons behind the fact, namely (1) the use of Google Scholar as a source of papers. It lags the time which Google Scholar needs to search papers against a query. (2) calculate feature values of candidate papers against every search query. The response time will be reduced if a local database is maintained in the system. Then, for every search query, it is not required to crawl Google Scholar as well as rerun the feature extraction process.
- The crawler used in SHARE is designed to crawl only 3 pages of Google Scholar to reduce the search time. The relevant papers may exist on other pages also. In this problem, the parallel crawler can be thought of as one of the possible solutions.
- Though SHARE has taken precautions against Google Scholar's crawler blocking, this has been done to a certain extent.
- The scalability issue is not considered in SHARE. SHARE can be implemented in a distributed environment to deal with situations when the quantity of articles and users has escalated.

6. Conclusions

This paper describes the implementation strategy of a content-based research paper recommendation system, SHARE. SHARE aims to address the shortcomings of existing recommendation systems along with the present research gap. The main contribution of SHARE lies in the process of article representation, user activity monitoring, and ranking method for representing recommendation lists. For the article representation, SHARE utilized extensive content analysis. In this regard, several features are identified to represent a paper. Further, to monitor user activity, the proposed system considers user's intention prediction to capture their dynamic notion. For ranking the recommendation list, SHARE proposed a novel ranking algorithm utilizing multiple criteria of papers along with the user's choice to make the recommendation personalized and improve the quality. In addition, SHARE is able to recommend old as well as new users. Finally, rigorous experiments have been carried out to evaluate the proposed system. The evaluation technique measures the novelty and diversity of SHARE together with relevancy, precision, and recall. Finally, the comparative analysis with the state-of-the-art systems proves the efficacy of the proposed SHARE.

References

[1] X. Gao, F. Feng, H. Huang, X.-L. Mao, T. Lan, Z. Chi, Food recommendation with graph convolutional network, Information Sciences 584 (2022) 170–183.
[2] A. Li, B. Yang, F. K. Hussain, H. Huo, Hsr: hyperbolic social recommender, Information Sciences 585 (2022) 275–288.
[3] M. Dhanda, V. Verma, Recommender system for academic literature with incremental dataset, Procedia Computer Science 89 (2016) 483–491.
[4] J. Lee, K. Lee, J. G. Kim, Personalized academic research paper recommendation system, arXiv preprint arXiv:1304.5457 (2013).
[5] C. Wang, D. M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 448–456.

[6] M. Alfarhood, J. Cheng, Collaborative attentive autoencoder for scientific article recommendation, in: 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2019, pp. 168–174.
[7] A. Mnih, R. R. Salakhutdinov, Probabilistic matrix factorization, Advances in neural information processing systems 20 (2007) 1257–1264.
[8] G. Wang, X. He, C. I. Ishuga, HAR-SI: A novel hybrid article recommendation approach integrating with social information in scientific social network, Knowledge-Based Systems 148 (2018) 85–99.
[9] D. Wang, Y. Chen, A novel cascade hybrid many-objective recommendation algorithm incorporating multistakeholder concerns, Information Sciences 577 (2021) 105–127.
[10] H. Liu, H. Kou, C. Yan, L. Qi, Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph, Complexity (2020). doi:https://doi.org/10.1155/2020/2085638.
[11] F. Xia, H. Liu, I. Lee, L. Cao, Scientific article recommendation: Exploiting common author relations and historical preferences, IEEE Transactions on Big Data 2 (2) (2016) 101–112.
[12] Z. Ali, G. Qi, K. Muhammad, B. Ali, W. A. Abro, Paper recommendation based on heterogeneous network embedding, Knowledge-Based Systems 210 (2020) 106438.
[13] O. Liadal, Explainable research paper recommendation using scientific knowledge graphs, Master's thesis (2021).
[14] J. K. Tarus, Z. Niu, A. Yousif, A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining, Future Generation Computer Systems 72 (2017) 37–48.
[15] H. Samin, T. Azim, Knowledge based recommender system for academia using machine learning: A case study on higher education landscape of pakistan, IEEE Access 7 (2019) 67081–67093.
[16] A. Kanakia, Z. Shen, D. Eide, K. Wang, A scalable hybrid research paper recommender system for microsoft academic, in: The world wide web conference, 2019, pp. 2893–2899. doi:https://doi.org/10.1145/3308558.3313700.
[17] T. Achakulvisut, D. E. Acuna, T. Ruangrong, K. Kording, Science concierge: A fast content-based recommendation system for scientific publications, PloS one 11 (7) (2016) e0158423.
[18] S. Amara, R. R. Subramanian, Collaborating personalized recommender system and content-based recommender system using textcorpus, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 2020, pp. 105–109.
[19] C. Musto, Enhanced vector space models for content-based recommender systems, in: Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 361–364.
[20] K. Hong, H. Jeon, C. Jeon, Personalized research paper recommendation system using keyword extraction based on userprofile, Journal of Convergence Information Technology 8 (16) (2013) 106.
[21] J. Sun, Y. Jiang, X. Cheng, W. Du, Y. Liu, J. Ma, A hybrid approach for article recommendation in research social networks, Journal of Information Science 44 (5) (2018) 696–711.
[22] B. Bulut, B. Kaya, R. Alhajj, M. Kaya, A paper recommendation system based on user's research interests, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 911–915.
[23] J. Gautam, E. Kumar, An improved framework for tag-based academic information sharing and recommendation system, in: Proceedings of the World Congress on Engineering (WCE'12), Vol. 2, London, U.K., 2012, pp. 1–6.
[24] B. Bulut, B. Kaya, M. Kaya, A paper recommendation system based on user interest and citations, in: 2019 1st International Informatics and Software Engineering Conference (UBMYK), IEEE, 2019, pp. 1–5.
[25] T. Chakraborty, A. Krishna, M. Singh, N. Ganguly, P. Goyal, A. Mukherjee, Ferosa: A faceted recommendation system for scientific articles, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2016, pp. 528–541.
[26] H. Liu, Z. Yang, I. Lee, Z. Xu, S. Yu, F. Xia, Car: Incorporating filtered citation relations for scientific article recommendation, in: 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), IEEE, 2015, pp. 513–518.
[27] A. Chaudhuri, D. Samanta, M. Sarma, Modeling user behaviour in research paper recommendation system, https://arxiv.org/abs/2107.07831 (2021).
[28] A. Chaudhuri, N. Sinhababu, M. Sarma, D. Samanta, Hidden features identification for designing an efficient research article recommendation system, International Journal on Digital Libraries 22 (2) (2021) 233–249.
[29] K. Chandrasekaran, S. Gauch, P. Lakkaraju, H. P. Luong, Concept-based document recommendations for citeseer authors, in: International conference on adaptive hypermedia and adaptive web-based systems, Springer, 2008, pp. 83–92.
[30] N. Yang, J. Jo, M. Jeon, W. Kim, J. Kang, Semantic and explainable research-related recommendation system based on semi-supervised methodology using bert and lda models, Expert Systems with Applications (2021) 116209.
[31] S. F. Chen, J. Goodman, An empirical study of smoothing techniques for language modeling, Computer Speech & Language 13 (4) (1999) 359–394.
[32] M. J. Pazzani, D. Billsus, Content-based recommendation systems, in: The adaptive web, Springer, 2007, pp. 325–341.

[33] T. Ebesu, Y. Fang, Neural citation network for context-aware citation recommendation, in: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, ACM, 2017, pp. 1093–1096.
[34] X. Cai, J. Han, L. Yang, Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018, pp. 5747–5754.
[35] Y. Xie, Y. Sun, E. Bertino, Learning domain semantics and cross-domain correlations for paper recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2021, pp. 706–715.
[36] J. Sun, J. Ma, Z. Liu, Y. Miao, Leveraging content and connections for scientific article recommendation in social computing contexts, The Computer Journal 57 (9) (2014) 1331–1342.
[37] L. Berkani, R. Hanifi, H. Dahmani, Hybrid recommendation of articles in scientific social networks using optimization and multiview clustering, in: International Conference on Smart Applications and Data Analysis, Springer, 2020, pp. 117–132.
[38] G. Wang, H.-R. Wang, Y. Yang, D.-L. Xu, J.-B. Yang, F. Yue, Group article recommendation based on er rule in scientific social networks, Applied Soft Computing 110 (2021) 107631.
[39] J. Zhang, L. Zhu, Citation recommendation using semantic representation of cited papers' relations and content, Expert Systems with Applications 187 (2022) 115826.
[40] M. Thushara, T. Mownika, R. Mangamuru, A comparative study on different keyword extraction algorithms, in: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), IEEE, 2019, pp. 969–973.
[41] A. Chaudhuri, D. Samanta, M. Sarma, Two-stage approach to feature set optimization for unsupervised dataset with heterogeneous attributes, Expert Systems with Applications 172 (2021) 114563.
[42] P. Wang, Z. Zhu, Y. Wang, A novel hybrid mcdm model combining the saw, topsis and gra methods based on experimental design, Information Sciences 345 (2016) 27–45.
[43] Y. Zhu, D. Tian, F. Yan, Effectiveness of entropy weight method in decision-making, Mathematical Problems in Engineering (2020).
[44] S. Guo, Application of entropy weight method in the evaluation of the road capacity of open area, in: AIP Conference Proceedings, Vol. 1839, AIP Publishing LLC, 2017, p. 020120.
[45] C.-P. Lee, C.-J. Lin, Large-scale linear ranksvm, Neural computation 26 (4) (2014) 781–817.
[46] P. Kaur, M. Singh, G. S. Josan, Comparative analysis of rank aggregation techniques for metasearch using genetic algorithm, Education and Information Technologies 22 (3) (2017) 965–983.
[47] K. Emamy, R. Cameron, Citeulike: a researcher's social bookmarking service, Ariadne (2007) 5.
[48] F. H. Del Olmo, E. Gaudioso, Evaluation of recommender systems: A new approach, Expert Systems with Applications 35 (3) (2008) 790–804.
[49] T. K. Kim, Understanding one-way anova using conceptual figures, Korean journal of anesthesiology 70 (1) (2017) 22.
[50] H. Abdi, L. J. Williams, Tukey's honestly significant difference (hsd) test, Encyclopedia of research design 3 (1) (2010) 1–5.