Received November 17, 2021, accepted November 26, 2021, date of publication November 30, 2021,
date of current version December 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3131470

Recommending Research Articles: A Multi-Level Chronological Learning-Based Approach Using Unsupervised Keyphrase Extraction and Lexical Similarity Calculation

TALHA BIN SARWAR[1], (Member, IEEE), NOORHUZAIMI MOHD NOOR[1], M. SAEF ULLAH MIAH[1], (Member, IEEE), MAMUNUR RASHID[2], (Member, IEEE), FAHMID AL FARID[3], (Member, IEEE), AND MOHD NIZAM HUSEN[4]

[1]Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pekan, Pahang 26600, Malaysia
[2]Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, Pekan, Pahang 26600, Malaysia
[3]Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63000, Malaysia
[4]Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Kuala Lumpur 50250, Malaysia

Corresponding author: Mohd Nizam Husen (mnizam@unikl.edu.my)

This work was supported in part by the Fundamental Research Grant Scheme (FRGS) under Grant FRGS/1/2019/ICT04-/UMP/02/1 (university reference RDU1901109), in part by the University Malaysia Pahang Flagship Grant under Grant RDU182201-3, and in part by Universiti Kuala Lumpur through the Short Term Research Grant under Grant str20017.

ABSTRACT A research article recommendation approach aims to recommend appropriate research articles to analogous researchers to help them better grasp a new topic in a particular research area. Due to the accessibility of research articles on the web, it is tedious to recommend a relevant article to a researcher who strives to understand a particular article. Most of the existing approaches for recommending research articles are metadata-based, citation-based, bibliographic coupling-based, content-based, and collaborative filtering-based. They require a large amount of data and do not recommend reference articles to the researcher who wants to understand a particular article going through the reference articles of that particular article. Therefore, an approach that can recommend reference articles for a given article is needed. In this paper, a new multi-level chronological learning-based approach is proposed for recommending research articles to understand the topics/concepts of an article in detail. The proposed method utilizes the TeKET keyphrase extraction technique, among other unsupervised techniques, which performs better in extracting keyphrases from the articles. Cosine and Jaccard similarity measures are employed to calculate the similarity between the parent article and its reference articles using the extracted keyphrases. The cosine similarity measure outperforms the Jaccard similarity measure for finding and recommending relevant articles to understand a particular article. The performance of the recommendation approach seems satisfactory, with an NDCG value of 0.87. The proposed approach can play an essential role alongside other existing approaches to recommend research articles.

INDEX TERMS Research article recommendation, chronological learning, keyphrase extraction, TeKET, cosine similarity, jaccard similarity.

I. INTRODUCTION

In recent years, research article recommendation systems have gained considerable attention in the research arena. One of the key reasons these recommendation systems get popular is that an automated system can recommend relevant research articles that the researchers/readers are looking for in the shortest period. On the other hand, the number of scientific research articles is increasing rapidly. Moreover, new articles are continuously being added to the scientific research field regularly. There are various subject areas or domains in research. According to Scopus, there are four research

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

domains: physical, health, social, and life sciences [1]. These domains also contain sub-domains, namely computer science, mathematics, material science, chemical engineering, medicine, social science, and many more. These different domains contain numerous research articles. It has been reported that there are about 25 million research articles available on the website [2]. The number is even larger today than it used to be. Therefore, finding a relevant research article for a particular article is a time-consuming and tedious task.

A research article contains concepts from different domains, subjects, or topics. Sometimes some topics are not discussed elaborately in an article. Instead, they are referred to the previously published articles. Therefore, to understand those topics, the researchers/readers read those referred articles. While searching for relevant research articles to understand a topic mentioned in a particular article, a researcher/reader usually needs to search for articles from the reference section. Finding a relevant article is time-consuming and a tedious task as one needs to read at least the title, abstract, or introduction to find whether the article is relevant or not. Therefore, the researchers/readers need to spend much time finding the relevant articles from the reference section to understand a particular topic of a particular article. However, the task of finding relevant research articles can be done quickly by introducing a new concept to research article recommendations that can recommend relevant reference articles for a particular article.

Most of the available recommendation approaches are researcher's profile-based [3]–[7] and recommend research articles as per user's research profiles and their past research histories. Some other recommendation approaches are query-based [8], [9] and return relevant articles based on the query from the researchers [10]. Apart from the user profile-based and user query-based approaches, there are some other approaches that includes collaborative filtering [8], [11], [12], content-based filtering [13], [14], citation-based [15]–[17], graph-based [18], and some other hybrid approaches [19], [20]. However, these approaches also incorporate the user/researcher profiles. Hence, no such work has yet been found to the best of our knowledge, which can recommend relevant reference articles for a particular article to understand. Therefore, this work proposes a new concept of research article recommendation approach that is a chronological learning-based [21] approach. Simple-to-complex, whole-to-part learning, prerequisite learning, and chronological learning are four traditional sequencing methods in the learning process [22]. The term "simple-to-complex" is self-explanatory. More uncomplicated principles are introduced first, followed by more nuanced ones in the learning process. This method of sequencing is used in many math textbooks. Whole-to-part learning gives learners a broad outline of a topic before delving further into specifics. In comparison to the inductive strategy of simple-to-complex learning, this is a deductive strategy. Prerequisite learning is a type of learning that requires basic knowledge to come before more advanced knowledge. It is similar to simple-to-complex instruction in that the order in which the prerequisite information is answered does not matter as much as enough of it is covered before moving on to the more complex knowledge. Finally, chronological learning occurs when the learning process is organized in such a way that the lessons/topics are taught in chronological order. For instance, history is a topic that, by definition, follows a chronological order. The theory of history is focused on the concept of chronology. The word "chronology" refers to the chronological order in which events took place. In the research area, the researchers also follow the chronological order in terms of publishing new articles. For example, a researcher published a new research article containing a new theory. In the future, other researchers might use that theory for solving many problems, and they cite that article in their articles. As a result, if a reader wants to read the newly published articles which contain that theory, he needs to read that cited article that contains the theory. A research article can cite various previously published articles, which a reader needs to read first to understand a new article in depth. Hence, a new recommendation approach is needed that can serve this kind of research article recommendation. Therefore, this proposed recommendation approach is termed chronological learning-based since it exploits the concept of chronological learning and utilizes the previously published research articles for the recommendation purpose that maintains chronological order. The chronological order means ordering the published research articles that are utilized as references in a particular article. This approach can recommend relevant reference articles to the researchers/readers to understand a particular article and gather more knowledge regarding different domains/subject areas in research.

This proposed approach can recommend research articles at different levels of a base/target article that researchers may be interested in. For example, a researcher may want to read and understand the concept of a research article. The article mentions various technical contexts or concepts that he may not understand. These contexts or concepts are not discussed in detail in this research article. Instead, they are being cited in this article. Therefore, the researcher must find the relevant articles in the references to help him understand the base article. This concept can be extended to multiple levels to understand the particular research article. For example, the base article has some reference articles in the reference section. Calculating the similarity of the reference articles with the base article to recommend relevant articles is considered as the first level of recommendation, and the base article is considered as the parent article for the first level. Again, each of the reference articles from the first level has its references. Each of the reference articles from the first level acts as a parent article in the second level. At the second level, relevant articles are recommended by calculating the similarity between the parent articles and the reference articles of these parent articles. Similarly, the following recommendation level can be called the third recommendation level. Thus, the proposed approach can be considered as a

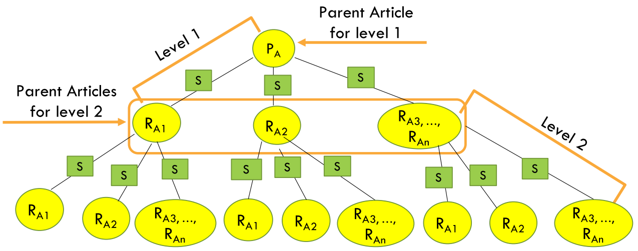

FIGURE 1. An example of level wise recommendation up to second level. Here, P_A denotes the parent article of first level. R_{A1} to R_{An} are the reference articles of parent article for each level. S denotes the similarity score between parent and reference articles. Each reference article of the first level act as the parent article for the second level recommendation.

multi-level approach. Figure 1 represents the concept of level-wise recommendation.

The proposed approach considers the full-text analysis of the articles for level-wise article recommendation since the full-text analysis is preferable for the document recommendation as it calculates the similarity between two documents [23]. The proposed approach computes the similarity of keywords of the articles. These words hold the key concepts of a particular article and can also be considered as keyphrases [24]. The proposed methodology employs TeKET [25], a tree-based unsupervised keyphrase extraction technique for extracting the keyphrases from articles. Though other widely utilized statistical-based and graph-based unsupervised keyphrase extraction techniques are available [26]–[28], TeKET performs better in extracting top quality keyphrases from research articles than the other techniques. Two widely utilized similarity calculation techniques named Cosine Similarity [29], and Jaccard Similarity [30] measures are employed to calculate the lexical similarity between two articles utilizing the extracted keyphrases. These two similarity calculation techniques tend to be the most effective and prominent techniques for similarity calculation. These techniques are widely utilized in the information retrieval or text processing tasks for similarity calculation [31], [32]. These techniques are also utilized in text similarity measures for news articles as well as combined with classifiers for text classification [33], [34]. Similarity scores are significant when it comes to recommending the most relevant research articles. In summary, the following are the major contributions of this work:

- A new concept of research article recommendation approach is proposed that utilizes the concept of chronological learning to recommend articles at multiple levels. This recommendation approach can help a researcher to understand a particular research article in depth by recommending research articles relevant to that article.
- A comparison is made between some prominent unsupervised keyphrase extraction approaches, namely tree-based, statistical-based, and graph-based, to find the suitable technique to extract keyphrases from the research articles.
- Lexical similarity is calculated employing Cosine Similarity and Jaccard Similarity measure utilizing the extracted keyphrases. Based on the similarity scores calculated between the parent and reference articles, the most relevant research articles are recommended.

The remainder of this article is structured in the subsequent manner. Section II discusses the related study. Section III discusses the problem formulation. Section IV describes the proposed methodology in detail. Section V describes the experimental details, result analysis, and discussion. Finally, section VI concludes the study with future directions.

II. RELATED STUDY

In recent times, many recommendation systems have been developed, however, quite a few of them are for recommending academic research articles [35]. These recommendation systems utilize various techniques, including content-based technique [36], collaborative filtering technique [37], content-based citation analysis [14], graph-based ranking algorithm [38], co-citation-based analysis [17], bibliographic coupling technique [10], [39], and many more.

The most state-of-the-art approach is the collaborative filtering approach for research article recommendation systems from all these approaches. However, this technique can create a data sparsity problem for a big citation matrix [8], [40]. Moreover, this approach cannot recommend articles to understand a particular article. Conversely, the content-based approaches [41]–[43] can overcome the mentioned disadvantages of collaborative filtering by comparing textual information between two articles [44]. However, this approach creates cold start problems for recommending new articles if a new article is not recommended previously to anyone. Additionally, the existing content-based approaches do not recommend relevant articles from a particular article to understand that.

In several similar analyses, content-based analysis is incorporated with co-citation analysis, and it gives better accuracy for research article recommendation [45]. Herein, co-citation analysis considers the relationship between two articles when a particular article cites them. On the other hand, the bibliographic coupling technique finds the relation between two articles when both the articles cite the same article [10], [39]. However, these approaches do not consider the text analysis and do not recommend relevant research articles to understand a particular article.

In [5] and [6], the user profile-based approaches recommend research articles to the researcher relevant to his field. These approaches work based on the researcher's previous works and citations. The user profiles are made based on the publications and their citations. That means these approaches also require a vast database to recommend a particular article to the relevant researcher. In several cases, it can give poor results if there is inadequate information. Moreover, these approaches cannot recommend relevant research articles since they are based on user profiles and do not consider the contextual text of the articles. In [18], a keyword-driven article recommendation approach is proposed utilizing an undirected paper citation graph for recommending research articles. The keywords taken from the user query are regarded as a Steiner tree problem. It also employs PP (Paper Popularity) method to find the optimal recommended papers. However, there can also be sparsity problems in the paper citation graph, and it recommends articles considering the citation analysis.

Some other hybrid approaches incorporate content-based filtering, collaborative filtering, or citation analysis to recommend research articles. In [20], an agent is created to recommend research articles to the users. This agent utilizes a hybrid approach since it incorporates content-based filtering and collaborative filtering. However, this is domain-dependent and recommends articles based on the user's interest. In [19], a hybrid approach is proposed utilizing the multilevel citation network and collaborative filtering. This approach is also author dependant and can give poor recommendations due to this dependency.

In [46], a web-based tool named Mendeley utilizes a profile-based technique to recommend a research article. Based on the researcher's last read article, it recommends whether from the desktop browser or mobile app. Moreover, it considers the researcher's reference list from the library and also his profile information. However, this recommendation process is not ideal as it works only based on previously saved data. Google Scholar is one of the most eminent recommendation systems which utilizes a statistical model for the research article recommendation based on co-authorships and citations [47]. However, they never disclose the statistical model they utilize for the recommendation. Google Scholar utilizes the author's profile for the recommendation based on that author's published articles by matching the indexes [47]. It recommends articles based on author profile data, research interests, citation of the article, and the research domains.

In [48], a tool named Action Science Explorer (ASE) is designed to help by searching keywords and finding relevant documents in return. This tool provides several features, like network visualization of research articles using citation analysis, text analysis by ''citing'' and ''cited by'' nature of articles, statistical methods, the summary of articles by analyzing ''in-cite'' text, making clusters and groups of the relevant articles. However, ASE needs a fixed dataset provided by the user. Therefore, an enhancement is made to ASE to automate the data collection process [49]. All the features are the same, just like before. However, in ASE, the relevant articles are ranked based on the citation analysis, citation count, and different metrics like the year of publication and so on. For ranking the papers, text similarity calculation may give a deep insight alongside the citation analysis. Moreover, there is no chronological order maintained in the ASE that differs from the prime concept of this work.

Since all the existing article recommendation systems mentioned above are based on profile-based, previous record-based, or citation-based, this work aims to propose a new concept to recommend reference articles for a particular article that enlightens the concept of chronological learning.

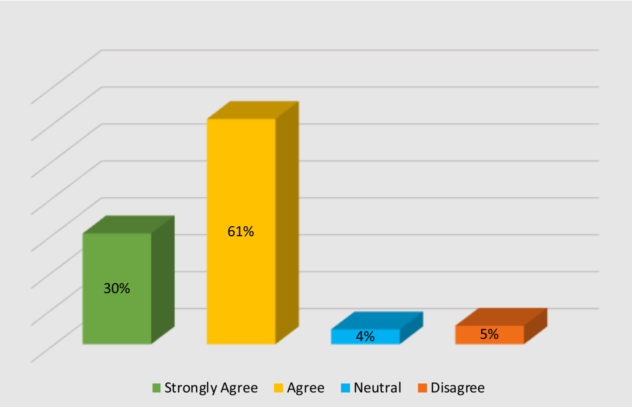

FIGURE 2. Preference of readers to read the reference articles for understanding an article.

III. PROBLEM FORMULATION

Since chronological learning utilizes level-wise article recommendation, for a base article B_A, up to three levels are considered in this work. The levels are selected up to three because of the opinion got from researchers from an expert survey. An expert survey is conducted upon more than a hundred researchers (professor, associate professor, assistant professor, lecturer, PhD student) worldwide. The prime concern of that survey is to know,

> Do the researchers read the reference articles of a particular article to understand that article?

The response of this concern from the survey is depicted in Figure 2. From the survey's response, it can be observed that most of the researchers want to read previous articles for a particular targeted article. The second concern of that survey is to know,

> Do the researchers read the reference articles of that reference articles? Up to how many levels do they read like this?

The response for this concern from the survey is depicted in Figure 3.

From the survey's response, it can be observed that the majority of the readers want to read up to three levels. Hence, this work recommends relevant articles to the researchers/readers for a targeted article up to three levels.

Let us consider an article that one wants to understand. We consider this article as a Base article B_A. This B_A is considered as the parent article P_A for level one. To understand this article's various terms and concept, we need to select the most relevant articles from the reference articles. Let us denote reference articles as $(R_A)s$ of P_A, $(R_A)s = \{R_{A1}, R_{A2}, R_{A3}, \ldots, R_{An}\}$, where $n \in Z+$. To find the best relevant article, we need to calculate the similarity between P_A and $(R_A)s$. For this, we need to extract keyphrases $(K_{P_A})s$ with their weights $(W_{K_{P_A}})s$ for P_A. Herein,

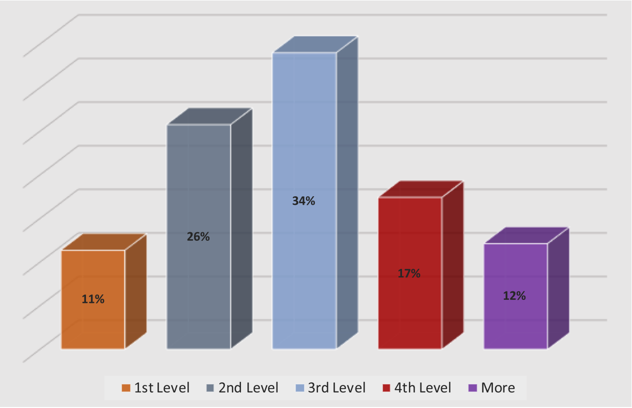

FIGURE 3. Preference of readers to read reference articles up to certain levels.

$K_{P_A} = \{K_{P_A}1, K_{P_A}2, K_{P_A}3, \ldots, K_{P_A}m\}$ where $m \in Z+$. Likewise, we need to extract keyphrases $(K_{R_A})s$ with their weights $(W_{K_{R_A}})s$ for each $R_A \in (R_A)s$. Herein, $(K_{R_A})s = \{K_{R_A(1,1)}, K_{R_A(1,2)}, K_{R_A(1,3)}, \ldots, K_{R_A(i,j)}\}$, where i, j denote the i-th article and j-th keyphrase respectively. After that, a new set of articles S_k can be found, sorted in descending order based on the similarity score calculated between P_A and $(R_A)s$. Herein, $S_k = \{R_{A1}, R_{A2}, R_{A3}, \ldots, R_{Ak}\}$, where $k \in Z+$. The top-5 articles with the highest similarity scores (A_{rec})s are recommended for P_A for that particular level. The top most article in (A_{rec})s is then considered as the P_A for the next level. The same thing is repeated up to the third level.

IV. PROPOSED METHODOLOGY

This section gives an extensive overview of the proposed methodology for the chronological learning-based research article recommendation approach for a given article. The methodology can be divided into four different phases; *i*) Data acquisition and processing, *ii*) keyphrase extraction, *iii*) similarity calculation and *iv*) recommending chronological learning-based articles. These four phases are depicted in figure 4 and elaborately discussed in the following subsequent sections.

A. DATA ACQUISITION AND PROCESSING

The discussion mentioned above states that the concept of chronological learning is a new one. Hence, no such dataset has been found so far as per our knowledge. Hence, for the proof of concept, a new dataset is prepared for experimental purposes. The research articles that are utilized for preparing the dataset are from the Computer Science domain. Initially, considering an article as B_A, all the reference articles $(R_A)s$ available online are acquired for further processing in level one. For all the $(R_A)s$ in level one, the same approach is performed in level two and level three as per the survey feedback mentioned in section III. The articles which are not found online are being omitted in this work. All the articles are scholarly articles from various journals and proceedings. All the articles in the dataset are in pdf format. Hence, one of the essential parts of this phase is converting the articles from pdf to text for further processing and similarity calculation. All the articles are converted from pdf to text. Then all the unnecessary stop words are removed. After that, the author provided keywords of the articles are separated from the text. These keywords are considered as the gold standard keyphrases since the authors provide them. These keywords are needed for the evaluation part for selecting the keyphrase extraction techniques in the next phase.

B. KEYPHRASE EXTRACTION

As mentioned earlier, there are millions of research articles from different domains in research; it is quite challenging to employ a supervised keyphrase extraction technique as different domains contain information that is almost different from each other. Additionally, supervised techniques need lots of training information. However, some domains lack training information. For this reason, the unsupervised technique is considered in this work for keyphrase extraction. The keyphrase extraction phase is crucial for the rest of the phases as it helps further for the similarity calculation and recommendation. Keyphrase extraction is one of the essential phases in this article recommendation approach as it helps calculate the similarity in the next phase. Without the proper keyphrase extraction technique, even the best recommendation system can perform poorly because of the poor similarity score calculated utilizing the extracted keyphrases [50]. Quality keyphrases help to summarize and identify a document well enough [25].

For the keyphrase extraction, TeKET is selected, which is a tree-based unsupervised keyphrase extraction technique [25]. TeKET is a domain-independent keyphrase extraction technique that utilizes limited statistical knowledge, and it requires no training data [25]. The process of extracting keyphrases utilizing TeKET can be divided into several phases, namely *i*) selection of candidate keyphrases, *ii*) processing of candidate keyphrases, and *iii*) selecting final keyphrases from candidate keyphrases. The functional details of TeKET are depicted in Figure 5.

1) CANDIDATE KEYPHRASE SELECTION

The candidate keyphrases are selected by employing Parts of Speech Tagging (POS Tagging) from the articles. Noun phrases are considered here as targeted candidate keyphrases since most of the time, the noun phrases are considered as candidate keyphrases [51]. The following POS pattern is employed for this purpose, which has been shown in [52] to be one of the most suitable patterns for extracting the most significant candidate keyphrases.

$$(< NN.* > + < JJ.* > ?)|(< JJ.* > ? < NN.* > +)$$

Here, NN denotes the nouns and JJ denotes the adjective. This is actually called regular expression and can be written by NLTK's (Natural Language Toolkit) RegexParser. Afterward, the most suitable candidate keyphrases are selected

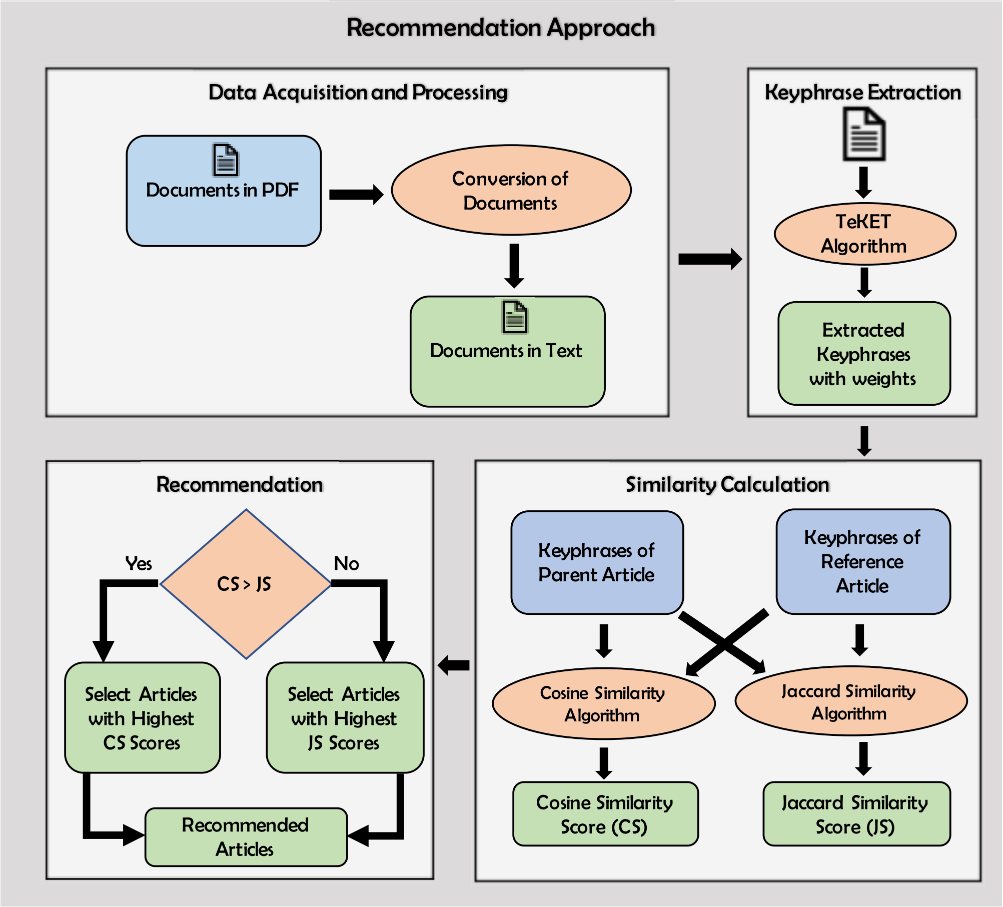
Recommendation Approach
Data Acquisition and Processing
Documents in PDF
Conversion of Documents
Documents in Text
Keyphrase Extraction
TeKET Algorithm
Extracted Keyphrases with weights
Recommendation
CS > JS
Yes
No
Select Articles with Highest CS Scores
Select Articles with Highest JS Scores
Recommended Articles
Similarity Calculation
Keyphrases of Parent Article
Keyphrases of Reference Article
Cosine Similarity Algorithm
Jaccard Similarity Algorithm
Cosine Similarity Score (CS)
Jaccard Similarity Score (JS)

FIGURE 4. **Different phases of research article recommendation approach. The first two phases, namely data acquisition & processing and keyphrase extraction are crucial for the last two phases, namely similarity calculation and recommendation.**

based on removing the words that contain a single alphabet or no alphabets and lower frequency words.

2) CANDIDATE KEYPHRASE PROCESSING USING KePhEx TREE

Candidate keyphrases are not processed in traditional unsupervised keyphrase extraction procedures; instead, they are delivered to the ranking step directly after selection. An intermediary step between candidate keyphrase selection and ranking might ease the unnecessary rating of extracted keyphrases, allowing for the extraction of more relevant keyphrases. After selecting the candidate keyphrases in TeKET, these are further processed utilizing a Keyphrase Extraction Tree (KePhEx) [52].

KePhEx is a binary tree, which can extract final keyphrases from candidate keyphrases. Although there are many other binary trees, the KePhEx tree is distinguished by the fact that both the position and the level of each node in the tree are fixed. Even a node's higher-level ancestors (including the root) are also fixed. The KePhEx tree starts building by selecting a root from candidate keyphrases as it can lead to forming a good tree for selecting the final keyphrases. Since noun phrases are the most likely candidates for final keyphrases, using them as roots enhances the likelihood of obtaining good final keyphrases. For the tree formation, after selecting a root, all the candidate keyphrases are selected that contain the root. These are considered as similar candidate keyphrases. The whole process for extracting the keyphrases follows three steps, namely *i*) tree formation, *ii*) tree processing, and *iii*) keyphrase extraction. The detailed algorithm for tree formation and tree processing can be found in [25].

In a KePhEx tree, each node contains a 2-tuple of data along with additional information, such as a word and its Cohesive Index (CI) value concerning the root. The CI has two advantages: *i*) it helps in determining the coherence of different words concerning the root of the tree, which is used as a ranking factor for keyphrase extraction, and *ii*) it provides flexibility in keyphrase extraction since the value of CI changes depending on the presence of the word in the candidate keyphrases. A good keyphrase must consist of a logically coherent set of words that occur regularly in the text.

The KePhEx tree follows three hypotheses mentioned in [25] to extract final keyphrases. Based on the hypotheses, the KePhEx tree extends, shrinks, or remains in the same

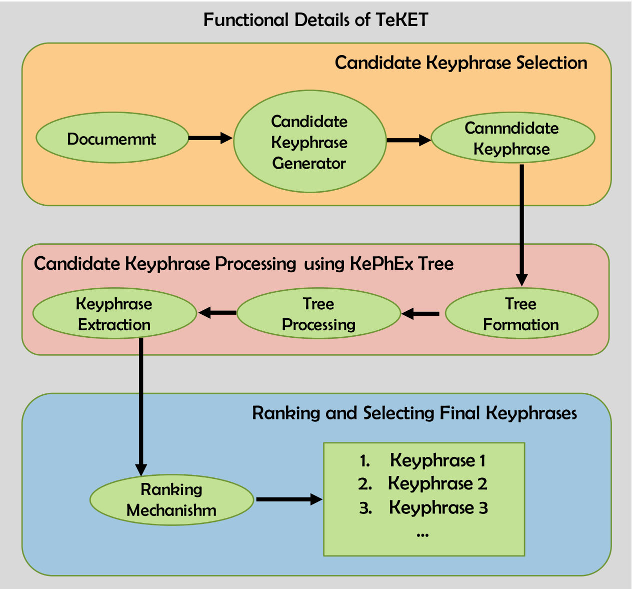
Functional Details of TeKET
Candidate Keyphrase Selection
Documemnt
Candidate Keyphrase Generator
Cannndidate Keyphrase
Candidate Keyphrase Processing using KePhEx Tree
Keyphrase Extraction
Tree Processing
Tree Formation
Ranking and Selecting Final Keyphrases
Ranking Mechanishm
1. Keyphrase 1
2. Keyphrase 2
3. Keyphrase 3
...

FIGURE 5. Functional overview of TeKET keyphrase extraction technique. In the very first step, Candidate keyphrases are selected from whole text of the article. Then these candidate keyphrases are processed using KePhEx Tree algorithm and finally they are ranked and selected as final keyphrases.

condition. For the keyphrase extraction, the weak nodes are pruned based on the CI value concerning the root. A threshold value called *minimum allowable μ (mamu)*, is utilized for this. A node with a lower value than the threshold gets removed from the tree. The *mamu* value is being updated periodically. The detailed algorithm for the pruning process and keyphrase extraction can be found in [25].

Hence, it can be said that using the KePhEx tree in keyphrase extraction has three advantages: *i*) it extracts high-quality keyphrases from candidate keyphrases, *ii*) it allows flexibility in keyphrase extraction, and *iii*) it helps to rank by providing a value representing the coherence of a word in a keyphrase concerning a root.

3) RANKING AND SELECTING FINAL KEYPHRASES

Generally, automatic keyphrase extraction techniques extract good quality keyphrases, yet extracting top quality keyphrases is necessary due to various applications like recommender systems or document clustering. TeKET employs a new keyphrase ranking approach utilizing the value CI or μ of a word concerning root in a keyphrase and Term Frequency (TF) to extract top-ranked keyphrases. To rank the final keyphrases, the weights are calculated utilizing the following Equation (1).

$$W_{kp} = \sum_{n=1}^{M} TF_n \times \sum_{n=1}^{M} \mu_n \tag{1}$$

Herein, W_{kp} is the calculated weight for a final keyphrase. M is the number of words in a keyphrase. TF is the term frequency and μ is the *mamu* value.

From that point onward, the TeKET extracts a good number of keyphrases from the candidate keyphrases. In this keyphrase extraction phase, the (K_{P_A})s from P_A and the (K_{R_A})s from $(R_A)s$ are extracted for the calculation of the next phase. TeKET also provides $(W_{K_{P_A}})$s and $(W_{K_{R_A}})$s for P_A and $(R_A)s$ respectively by employing Equation (1).

C. SIMILARITY CALCULATION

Similarity calculation is another primary concern of this proposed methodology as it calculates the similarities between P_A and $(R_A)s$ at each level. After extracting the necessary keyphrases from P_A and $(R_A)s$ in each level, similarities are calculated between the P_A and $(R_A)s$. The lexical similarity calculation approach is employed to determine the similarity between the P_A and $(R_A)s$. Lexical similarity computation is a measure for comparing the similarity between two texts, which is based on the intersection of word sets acquired from the texts [53]. Two different similarity calculation techniques are employed. One is Cosine Similarity (CS) [29], and the another one is Jaccard Similarity (JS) [30]. Both the similarity calculation techniques are widely known for similarity calculation between different documents. The similarity is calculated utilizing the extracted keyphrases from the articles.

1) COSINE SIMILARITY

Cosine Similarity (CS) is well known as a similarity measure index which is being utilized widely is based on Euclidean distance [29]. Cosine Similarity is employed to measure the similarity between various documents. The Cosine Similarity calculates the distance between two vectors by computing the angle ($cos(\theta)$) utilizing a dot product. Even though the lengths of the two papers differ significantly, there is still a chance that they would be similar due to the smaller angle, which corresponds to a greater similarity score. It is also mentioned that Cosine Similarity can check similarity semantically [54]. Here, CS between P_A and $(R_A)s$ can be calculated by the following Equation (2).

$$CS(P_A, R_A) = \frac{\sum_{i=1}^{n} W_{K_{P_{Ai}}} W_{K_{R_{Ai}}}}{\sqrt{\sum_{i=1}^{n} W^2_{K_{P_{Ai}}}} \sqrt{\sum_{i=1}^{n} W^2_{K_{R_{Ai}}}}} \tag{2}$$

Herein, P_A is the parent article, and R_A is the reference article of the P_A. $W_{K_{P_{Ai}}}$ and $W_{K_{R_{Ai}}}$ be the weights of the i-th keyphrase in P_A and R_A, respectively. Here, the value of CS can vary between 0 and 1 based on the similarity between the two articles.

2) JACCARD SIMILARITY

Jaccard Similarity (JS) is a well-known similarity calculation index that performs well for the recommendation purpose than other similarity calculation models [30]. Jaccard Similarity usually compares two sets of words and finds similarities by calculating which data are distinct and shared. The intersection of (K_{P_A})s and (K_{R_A})s that is divided by the union of K_{P_A} and K_{R_A} is known as Jaccard Similarity. Here, JS between P_A and $(R_A)s$ can be calculated by the following

Equation (3).

$$JS(P_A, R_A) = \frac{|K_{P_A} \cap K_{R_A}|}{|K_{P_A} \cup K_{R_A}|} \tag{3}$$

Herein, P_A is the parent article, and R_A is the reference article of the P_A. K_{P_A} and K_{R_A} be the set of keyphrases of P_A and R_A, respectively. Here, the value of JS can vary between 0 and 1 based on the similarity between the two articles. The higher the value of JS, the articles are more similar. However, it may give a poor outcome due to having small-sized documents with fewer keyphrases. However, it is not an issue in the case of research articles since the documents are large enough to generate a good score.

D. RECOMMENDING CHRONOLOGICAL LEARNING-BASED ARTICLES

As aforementioned, in chronological learning, the similarity calculation starts at level one and continues up to level three as per the recommendation of the experts mentioned in the survey. This methodology recommends the top-5 $(R_A)s$ article in each level selected based on the scores calculated by the two different similarity measure techniques named Cosine and Jaccard Similarity. In each level, the value of CS and JS are calculated for each $R_A \in (R_A)s$ concerning to P_A. The process of recommending an article in each level is shown in Algorithm 1.

The computational complexity of the proposed approach for recommending articles in each level can be computed by evaluating the computational time of Algorithm 1. The computation of the time complexity of the algorithm is essential in this regard. The time taken by an algorithm to execute, as a function of the length of the input, is called the time complexity. Big-O notation is a measure for determining the complexity of an algorithm. It calculates how long each code statement in an algorithm takes to execute. In short, Big-O notation refers to the connection between the algorithm's input and the steps required to execute it. In this approach, The inputs are the parent and reference articles. Two articles are compared at the same time are a parent and a reference article. The time complexity for extracting keyphrases utilizing TeKET is $O(n^2)$. For both Cosine and Jaccard similarity calculation, the time complexity is $O(n^2)$. The other statements in Algorithm 1 have time complexity of $O(n)$ and $O(1)$. The overall complexity of this proposed approach would be $O(1) + O(n) + O(n^2) = O(n^2)$. Hence, the total time needed for the proposed approach is the quadratic time.

Algorithm 1: Recommendation of Articles in Each Level

```
Input: parent article ($P_A$), reference articles of parent
       article ($R_A$)s
Output: recommended articles ($A_{rec}$)s
/* All the notations utilize below
   are described in section III                    */
Select $P_A$
initialize $CS_{List} \leftarrow$ NULL
initialize $JS_{List} \leftarrow$ NULL
extract ($K_{P_A}$)s along with ($W_{K_{P_A}}$)s from $P_A$ utilizing
  TeKET
for $\forall R_A \in (R_A)s$ do
    extract ($K_{R_A}$)s and ($W_{K_{R_A}}$)s from $R_A$ utilizing TeKET
    calculate $CS$ utilizing Equation (2), employing
      ($W_{K_{P_A}}$)s and ($W_{K_{R_A}}$)s
    make a tuple, $t_{cs}$ utilizing ($articleName$, $CS$)
    append $t_{cs}$ in $CS_{List}$
    calculate $JS$ utilizing Equation (3), employing
      ($K_{P_A}$)s and ($K_{R_A}$)s
    make a tuple, $t_{js}$ utilizing ($articleName$, $JS$)
    append $t_{js}$ in $JS_{List}$
end for
Compare score of $CS_{List}$ and $JS_{List}$ for every $R_A$
append top-5 ($R_A$)s in ($A_{rec}$)s
return ($A_{rec}$)s
```

V. EXPERIMENTAL DETAILS, RESULT ANALYSIS, AND DISCUSSION

An experiment and subsequent analysis are carried out in order to assess the proposed methodology. The experimental details, result analysis, and discussion are extensively discussed in section V-A, section V-B, and section V-C, respectively.

A. EXPERIMENTAL DETAILS

This section provides an extensive overview of the experimental setup and the evaluation metrics utilized to evaluate the performance of the keyphrase extraction techniques and recommendation approach. The experimental setup and evaluation metrics are represented in section V-A1 and section V-A2, respectively.

1) EXPERIMENTAL SETUP

The proposed methodology is implemented employing the python programming language. Python 3.7 is utilized as the version. Several python packages are utilized such as stopwords, word_tokenize, sent_tokenize of Natural Language Toolkit (NLTK) [55] as well as other relevant packages like math [56] and os [57]. For the conversion of pdf to text, a library called Tika [58] is utilized. For the implementation of the statistical-based and graph-based approaches, the python keyphrase extraction toolkit (pke) [59] is utilized. For the tree-based approach, TeKET [60] is implemented. All the experimental codes and dataset are in [61] and will be provided upon request.

2) EVALUATION METRICS

Since the extracted keyphrases perform the most significant role in the similarity calculation part, it is essential to evaluate all the keyphrase extraction techniques in terms of their performance. This evaluation is accomplished by comparing the extracted keyphrases list with the gold standard keyphrase list provided by the authors of the articles. To assess the effectiveness of several unsupervised keyphrase extraction techniques, precision (ρ), recall (ϱ), and F1-score (δ) are calculated. These are very well-known evaluation metrics. ρ is the ratio of correctly extracted keyphrases over total extracted keyphrases. The ρ score reflects how well the keyphrase extraction techniques can accurately extract keyphrases within total extracted keyphrases. The following Equation (4) can be utilized to calculate ρ.

$$\rho = \frac{KP_{correct}}{KP_{extract}} \tag{4}$$

Herein, $KP_{correct}$ is the list of accurately matched keyphrases in an article with the gold standard keyphrases. On the other hand, $KP_{extract}$ is the list of total extracted keyphrases in an article.

ϱ, on the other hand, is the ratio of accurately extracted keyphrases over the actual gold standard keyphrases. The ϱ score reflects how well the keyphrase extraction technique can accurately extract keyphrases concerning the gold standard keyphrases. The following Equation (5) can be utilized to calculate ϱ.

$$\varrho = \frac{KP_{correct}}{KP_{gstandard}} \tag{5}$$

Herein, $KP_{correct}$ is the list of accurately matched keyphrases in an article with the gold standard keyphrases. On the other hand, $KP_{gstandard}$ is the list of total gold standard keyphrases in an article.

The score of ρ and ϱ is correlated in finding the overall performance of the proposed approach. The weighted average of ρ and ϱ is named the δ. δ is combining both the ρ and ϱ into a single measure. This combined measure provides a glimpse of the overall performance of the unsupervised keyphrase extraction techniques. The following Equation (6) can be utilized to calculate δ.

$$\delta = \frac{2 \times \rho \times \varrho}{\rho + \varrho} \tag{6}$$

This δ takes both false positives and false negatives into consideration. The δ is typically more important than accuracy when there is uneven data distribution.

The proposed chronological learning-based research article recommendation approach is evaluated employing the recommendation system evaluation metric, namely Normalized Discounted Cumulative Gain (NDCG) [62]. This metric is extensively utilized for the evaluation of recommendation systems [63]. Hence, it is employed in this work to measure the performance of the ranked recommended article list, A_{rec}. NDCG is the weighted average of the ranked relevance of recommended $(R_A)s$ of P_A. It determines how close the ranked recommended $(R_A)s$ are to the definitive ranking of the articles. The value of NDCG is calculated utilizing the following Equation (7).

$$NDCG_k = \frac{DCG_k}{IDCG_k} \tag{7}$$

Herein, the normalized gain accumulated at a certain rank k is denoted by $NDCG_k$. DCG_k denotes the total discounted cumulative gain at particular rank k for the recommended $(R_A)s$. On the other hand, $IDCG_k$ is the total ideal discounted cumulative gain at particular rank k, which is a DCG measure and denotes the best-ranked recommended articles [64]. The value of NDCG generally normalizes the value of DCG by dividing by the value of IDCG. The range of the NDCG value lies between 0 to 1. The NDCG value 1 denotes the perfect recommendation by the system. The DCG/IDCG can be calculated utilizing the following Equation (8).

$$DCG_k/IDCG_k = \sum_{j=1}^{k} \frac{rel_j}{\log_2(j+1)} \tag{8}$$

Herein, rel_j is the relevancy score at position j for the recommended R_A with respect to the P_A.

B. RESULT ANALYSIS

There exist various unsupervised keyphrase extraction techniques such as tree-based, statistical-based, and graph-based; it is intricate to choose one technique since the recommendation process depends much on the extracted keyphrases to help the further calculation of similarity indexes on the next phase. Hence, it is imperative to extract relevant keyphrases so that they can be utilized to calculate similarity and generate a more precise recommendation.

Several unsupervised keyphrase extraction techniques are employed in this work to examine which one works better for the keyphrase extraction. Two unsupervised keyphrase extraction techniques are employed from various statistical-based techniques; namely, YAKE [65], and Term Frequency and Inverse Document Frequency (TF-IDF) [66]. From various graph-based techniques, TopicRank (TR) [67], PositionRank (PR) [68], SingleRank (SR) [69], and MultipartileRank (MR) [70] unsupervised keyphrase extraction techniques are employed. Amid tree-based techniques, TeKET is employed, which is also an unsupervised keyphrase extraction technique. All these techniques are employed upon randomly selected articles from the dataset. After extracting the keyphrases, those are compared with the gold standard keyphrase list to measure the performance of the employed keyphrase extraction techniques. The gold standard keyphrase list is prepared with the author-provided keywords of those articles. The ρ, ϱ, and δ are calculated for the top-5, top-10, and top-15 extracted keyphrases, employing Equation (4), (5),and (6) respectively, to assess the performances of the employed unsupervised keyphrase extraction techniques. The performance of the employed unsupervised keyphrase extraction techniques is depicted in Table 1.

From Table 1, it can be evident that the tree-based unsupervised keyphrase extraction technique outperforms all the

TABLE 1. Performance of the employed unsupervised keyphrase extraction techniques for extracting top-5, top-10, and top-15 keyphrases.

Approach	Technique	Top-5			Top-10			Top-15		
		P	R	F1	P	R	F1	P	R	F1
Tree-base	TeKET	14.67	14.56	14.39	12.67	24.31	16.37	10.22	29.03	14.91
Statistical-base	YAKE	14.67	14.03	1.12	11.00	21.82	14.42	7.55	22.65	11.20
	TF-IDF	14.00	13.91	13.75	11.00	21.16	14.26	9.11	27.25	13.47
Graph-base	TR	8.67	8.55	8.44	7.00	12.9	8.93	6.22	16.7	8.94
	PR	14.66	14.19	14.26	9.33	17.95	12.14	6.44	18.5	9.47
	SR	0.67	0.48	0.56	1.33	2.64	1.75	1.78	4.98	2.60
	MR	12.00	11.18	11.32	9.33	17.66	12.01	8.22	23.11	11.95

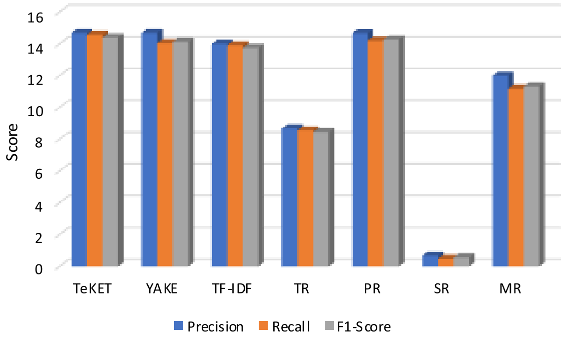

FIGURE 6. ρ, ϱ and δ for the Top-5 keyphrases extraction employing various unsupervised keyphrase extraction techniques.

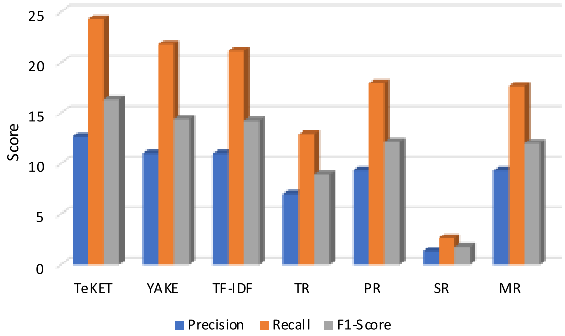

FIGURE 7. ρ, ϱ and δ for the Top-10 keyphrases extraction employing various unsupervised keyphrase extraction techniques.

other graph-based and statistical-based techniques. It produces the highest ρ value of 14.67, 12.67, and 10.22 for the top-5, top-10, and top-15, extracted keyphrases, respectively. These ρ values indicate that TeKET can extract more quality keyphrases among the other employed techniques. TeKET also cuts the highest ϱ value of 14.56, 24.31, and 29.03 for the top-5, top-10, and top-15, respectively, indicating that it can extract more accurate keyphrases concerning the gold standard keyphrases. δ shows the balancing between ρ and ϱ, which also considers the extracted keyphrases that are incorrect. TeKET cuts the highest δ of 14.39, 16.37, and 14.91 for the top-5, top-10, and top-15, respectively. The performance of the employed techniques for top-5, top-10, and top-15 extracted keyphrases are graphically represented in Figure 6, Figure 7, and Figure 8, respectively to give a better insight.

The overall performance of the employed unsupervised keyphrase extraction techniques can be depicted in Figure 9. From the Figure 9; it is evident that TeKET, which is a tree-based technique, outperforms all other employed statistical-based and graph-based techniques in terms of ρ and ϱ value. The δ line goes high for the TeKET. Hence, TeKET is selected as the keyphrase extraction technique in this work.

After selecting TeKET as the keyphrase extraction technique, the B_A and reference articles of that B_A, $(R_A)s$ are selected from the dataset for level one. In level one, the B_A is considered as the P_A of that level. After that, the keyphrases $(K_{P_A})s$ for P_A and $(K_{R_A})s$ for each R_A are extracted along with their weights. Then, the similarity between the P_A and $(R_A)s$ are calculated utilizing the Cosine similarity measure. After calculating CS, the top-5 articles with the highest similarity scores are recommended for level two. Therefore, each recommended article in level one is considered as the P_A for level two. The $(R_A)s$ are the reference articles of P_A for level two calculation. Likewise, in level one, the similarity calculation is done in levels two and three for recommending the most relevant articles in each level. To determine whether or not these recommended articles are relevant to the BA, CS is calculated between the BA and the $(A_{rec})s$ recommended at each level. The top-1 recommended articles in each level that is acquired employing the Cosine Similarity technique are depicted in Table 2.

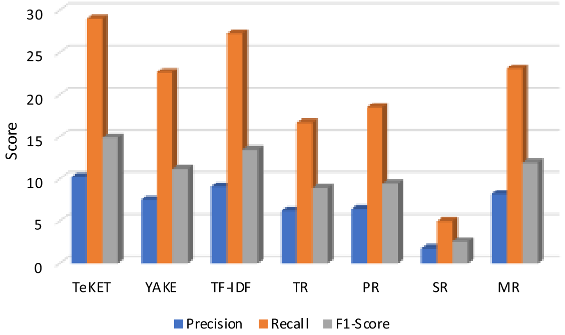

FIGURE 8. ρ, ϱ and δ for the Top-15 keyphrases extraction employing various unsupervised keyphrase extraction techniques.

TABLE 2. Recommending top-1 article in each level based on cosine similarity.

Level	Base Article Name	Parent Article Name	Recommended Article	CS of A_{rec} with P_A	CS of A_{rec} with B_A
Level 1	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"Unsupervised Keyphrase Extraction: Introducing New Kinds of Words to Keyphrases"	0.6679	0.6679
Level 2	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"Unsupervised Keyphrase Extraction: Introducing New Kinds of Words to Keyphrases"	"Clustering to Find Exemplar Terms for Keyphrase Extraction"	0.6450	0.5149
Level 3	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"Clustering to Find Exemplar Terms for Keyphrase Extraction"	"Learning to Extract Keyphrases from Text"	0.5368	0.5416

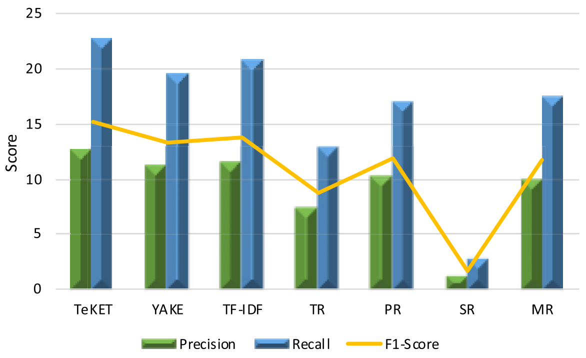

FIGURE 9. The overall performance of the employed various unsupervised keyphrase extraction techniques in terms of ρ, ϱ and δ.

Similarly, JS is also calculated up to three levels between P_A and $(R_A)s$. The top-5 articles with the highest similarity scores in each level are recommended. Again JS is calculated between the B_A and the recommended articles acquired in each level. The top-1 recommended articles in each level that is acquired employing the Jaccard Similarity technique are depicted in Table 3.

From Table 2 and Table 3, it is evident that the CS measure generates a better similarity score than the JS measure. It can also be observed that the recommended articles (A_{rec})s up to level three produce more likely similar scores with B_A as they produce with their P_A in each level.

C. DISCUSSION

As discussed earlier, utilizing the concept of chronological learning-based research article recommendation is a new approach; no prior work is found to the best of our knowledge, which recommends articles level-wise in a chronological manner to understand a particular article. A lexical similarity calculation approach is performed in this work to calculate the similarity between P_A and $(R_A)s$ in each level to recommend the most relevant articles. Initially, TeKET is employed, which is one of the core parts of this work to extract the keyphrases from articles. After that, CS and JS are calculated. Although this type of work is not yet available, there are some other works that are user profile-based collaborative research article recommendation approaches that utilize the traditional TF-IDF technique to extract keyphrases and rank them for similarity calculation [5], [9], [71], [72]. However, from Figure 9 it can be observed that TeKET easily outperforms TF-IDF as well as other keyphrase extraction techniques in terms of performance evaluation scores ρ, ϱ, and δ. TeKET has some advantages over other techniques since it is domain and language independent [25]. Moreover, the additional ranking mechanism of TeKET utilizing the CI adds additional flexibility to extract more quality keyphrases than the other techniques. The CS measure comparatively generates more similarity scores than the JS measure, which is observable from the similarity calculation approaches. The CS measure is advantageous since it can generate a smaller angle for producing the higher similarity score though two articles are apart by more Euclidean distance.

Generally, article recommendation systems can be examined by user study [73]. Performing a user study to evaluate a recommendation approach can be an effective way since it can give a view of the real-time performance of the proposed approach [12]. Since there is no benchmark approach regarding this work, the performance evaluation of the proposed chronological learning-based article recommendation approach is performed employing a user study performed by experts from the Computer Science domain. The experts manually rank the recommended articles by the CS and JS measures. The ranks made by the experts are considered the ground truth for evaluating the performance of the proposed approach. The ranking is made considering the relevancy score of the recommended $(R_A)s$ with P_A. The relevancy scores are classified into four categories and are depicted in Table 4.

Based on the relevancy score of the expert ranked articles, the NDCG values are calculated for the top-5 recommended articles (A_{rec})s by the CS and JS measures in each level by employing Equation (7). Figure 10 depicts the performance of the proposed recommendation approach in terms of NDCG values for the top-5 recommended articles by the CS and JS measures.

In Figure 10, the X-axis represents the levels up to three for recommending articles. The Y-axis represents the calculated NDCG values for both CS and JS. The NDCG values depict that the recommended top-5 articles by CS measure are more relevant for the targeted article B_A than the recommended top-5 articles by JS measure in each level. The recommendation made by the CS measure outperforms the JS measure

TABLE 3. Recommending top-1 article in each level based on Jaccard similarity.

Level	Base Article Name	Parent Article Name	Recommended Article	JS of A_{rec} with P_A	JS of A_{rec} with B_A
Level 1	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"Corpus-independent Generic Keyphrase Extraction Using Word Embedding Vectors"	0.3152	0.3152
Level 2	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"Corpus-independent Generic Keyphrase Extraction Using Word Embedding Vectors"	"Automatic Keyphrase Extraction: A Survey of the State of the Art"	0.2647	0.2679
Level 3	"PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents"	"Automatic Keyphrase Extraction: A Survey of the State of the Art"	"Automatic keyphrase extraction from scientific articles"	0.2515	0.2540

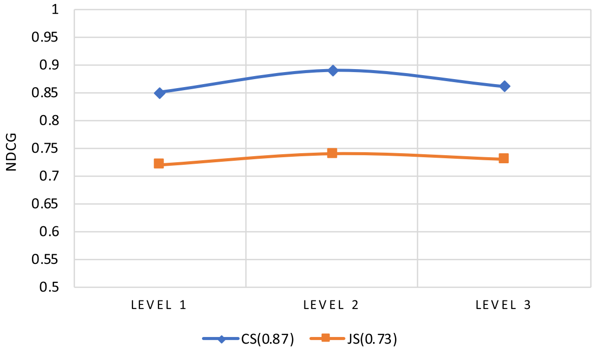

FIGURE 10. Performance of the proposed recommendation approach for both CS and JS. The NDCG value shows the relevancy of the recommended research articles for the particular targeted article.

TABLE 4. Relevancy categories with scores for ranking the recommended articles.

Category	Relevancy Score
Not Relevant	0
Somewhat Relevant	1
Relevant	2
Completely Relevant	3

with the NDCG values of 0.85, 0.89, and 0.86 for level one, level two, and level three, respectively, for recommending relevant articles. The overall NDCG value for the CS and JS measures in recommending research articles are 0.87 and 0.73, respectively.

For the more admissibility of the proposed chronological learning-based approach, another comparison is made in each level between the top-5 expert-recommended (ER) articles, and the CS measure recommended articles. For this, the similarity between the top-5 recommended articles and the targeted B_A are calculated in each level. Since the CS produces a more accurate recommendation, the similarity of ER articles with B_A are calculated employing the CS measure in each level and labeled as ERCS. The comparison is depicted in Figure 11.

In Figure 11, the X-axis represents the produced similarity score of the top-5 recommended articles by employing CS measure and ERCS. On the other hand, the Y-axis represents CS measure and ERCS in each level. It can be observed from

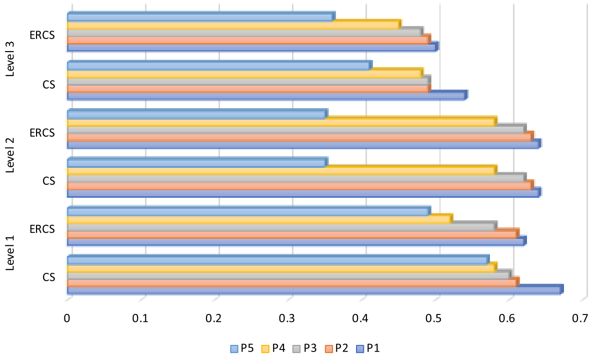

FIGURE 11. Comparison between CS and ERCS for recommending articles in each level. This figure shows the similarity score of the recommended articles using proposed approach and expert recommendation by calculating cosine similarity.

Figure 11 that the top-5 recommended articles by ERCS and CS measures are almost similar. Another notable observation can be found from Figure 11 that the top-5 recommended articles in level one are more similar to the B_A that the researcher/reader wants to understand compared to the recommended articles of level two and level three. This scenario seems logical since the level one reference articles $(R_A)s$ are the directly referred articles by B_A. Hence, it can be said that the proposed approach can recommend good quality research articles for a particular targeted article using the Keyphrase extraction technique TeKET and CS measure. However, this proposed approach may not be suitable for other text recommendation purposes, for instance, news articles. The reason behind this can be the writing style of the news article, which differs from the research articles. Moreover, the words of a research article are more cohesive than the news article.

VI. CONCLUSION AND FUTURE WORK

The utilization of a research article recommendation system in research has become a pressing challenge for handling a significant amount of articles and recommending relevant articles as per the reader's interest. The cutting-edge Google Scholar and other existing methodologies recommend research articles, yet they have drawbacks as they cannot recommend the level-wise relevant articles to understand a particular article. Moreover, the quality of these systems falls as citation count, and lack of content analysis of the articles are the major factors of these systems.

In this study, a new chronological learning-based research article recommendation approach is proposed to recommend previous articles mentioned in the reference of a particular article for understanding that article. The proposed methodology introduces a tree-based keyphrase extraction technique named TeKET, which can extract keyphrases from the articles for similarity calculation between parent and reference articles. The benefit of utilizing this TeKET is, it ranks the keyphrases utilizing the CI of that keyphrase in the article. Two different similarity calculation measures, namely Cosine and Jaccard Similarity, are employed to calculate the similarity between parent and reference articles at different levels. Afterward, outputs of the two similarity measures are compared to recommend the articles relevant to the base article. In this case, the Cosine Similarity measure outperforms the Jaccard Similarity measure with the overall NDCG value of 0.87. Therefore, the most similar articles utilizing the Cosine Similarity measure are considered for the recommendation.

The experiment in this study is performed upon a small dataset prepared for the proof of concept that includes research articles from the domain of computer science since no existing dataset is found. The result would have been more diverse if articles from the other domains had also been added. In the future, the experiment will be performed upon a large dataset having articles from different domains. On the other hand, the similarities are calculated upon the full text of the articles. Therefore, section-based similarity will be calculated, giving more insight to this work in finding relevant research articles. Furthermore, since lexical similarities are only calculated in this study, semantic similarity calculation between articles, a deep learning approach, might give a better recommendation for finding more relevant reference articles of a particular research article.

ACKNOWLEDGMENT

The authors would also like to thank Saiful Azad, a Professor and the Dean (Acting) of the Faculty of Science and Engineering, Green University of Bangladesh (GUB), and Gollam Rabby, a Research Assistant at the University of Economics, Prague, for helping with the survey work.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] A. Joshi, "Comparison between Scopus and ISI web of science," *J. Global Values*, vol. 7, no. 1, pp. 1–11, 2016.

[2] M. Khabsa and C. L. Giles, "The number of scholarly documents on the public web," *PLoS ONE*, vol. 9, no. 5, May 2014, Art. no. e93949.

[3] X. Tang and Q. Zeng, "Keyword clustering for user interest profiling refinement within paper recommender systems," *J. Syst. Softw.*, vol. 85, no. 1, pp. 87–101, Jan. 2012.

[4] D. De Nart, F. Ferrara, and C. Tasso, "Personalized access to scientific publications: From recommendation to explanation," in *Proc. Int. Conf. User Modeling, Adaptation, Personalization*. Berlin, Germany: Springer, 2013, pp. 296–301.

[5] K. Sugiyama and M.-Y. Kan, "Scholarly paper recommendation via user's recent research interests," in *Proc. 10th Annu. Conf. Digit. Libraries (JCDL)*, 2010, pp. 29–38.

[6] B. Kaya, "User profile based paper recommendation system," *Int. J. Intell. Syst. Appl. Eng.*, vol. 6, no. 2, pp. 151–157, 2018.

[7] H. Xue, J. Guo, Y. Lan, and L. Cao, "Personalized paper recommendation in online social scholar system," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 612–619.

[8] A. E. S. Tafreshi, A. S. Tafreshi, and A. L. Ralescu, "Ranking based on collaborative feature weighting applied to the recommendation of research papers," *Int. J. Artif. Intell. Appl.*, vol. 9, no. 2, pp. 47–53, Mar. 2018.

[9] C. Nascimento, A. H. F. Laender, A. S. da Silva, and M. A. Gonçalves, "A source independent framework for research paper recommendation," in *Proc. 11th Annu. Int. ACM/IEEE Conf. Digit. Libraries (JCDL)*, 2011, pp. 297–306.

[10] R. Habib and M. T. Afzal, "Sections-based bibliographic coupling for research paper recommendation," *Scientometrics*, vol. 119, no. 2, pp. 643–656, 2019.

[11] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, "On the recommending of citations for research papers," in *Proc. ACM Conf. Comput. Cooperat. Work (CSCW)*, 2002, pp. 116–125.

[12] K. Sugiyama and M.-Y. Kan, "Exploiting potential citation papers in scholarly paper recommendation," in *Proc. 13th ACM/IEEE-CS Conf. Digit. Libraries (JCDL)*, 2013, pp. 153–162.

[13] N. Ratprasartporn and G. Ozsoyoglu, "Finding related papers in literature digital libraries," in *Proc. Int. Conf. Theory Pract. Digit. Libraries*. Berlin, Germany: Springer, 2007, pp. 271–284.

[14] Y. Ding, G. Zhang, T. Chambers, M. Song, X. Wang, and C. Zhai, "Content-based citation analysis: The next generation of citation analysis," *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 9, pp. 1820–1833, 2014.

[15] N. J. van Eck and L. Waltman, "Citation-based clustering of publications using CitNetExplorer and VOSviewer," *Scientometrics*, vol. 111, no. 2, pp. 1053–1070, 2017.

[16] D. Yu, Z. Xu, W. Pedrycz, and W. Wang, "Information sciences 1968–2016: A retrospective analysis with text mining and bibliometric," *Inf. Sci.*, vols. 418–419, pp. 619–634, Dec. 2017.

[17] X. Y. Leung, J. Sun, and B. Bai, "Bibliometrics of social media research: A co-citation and co-word analysis," *Int. J. Hospitality Manage.*, vol. 66, pp. 35–45, Sep. 2017.

[18] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, pp. 1–15, Apr. 2020.

[19] W. Waheed, M. Imran, B. Raza, A. K. Malik, and H. A. Khattak, "A hybrid approach toward research paper recommendation using centrality measures and author ranking," *IEEE Access*, vol. 7, pp. 33145–33158, 2019.

[20] K. Chekima, C. K. On, R. Alfred, and P. Anthony, "Document recommender agent based on hybrid approach," *Int. J. Mach. Learn. Comput.*, vol. 4, no. 2, pp. 151–156, Apr. 2014.

[21] O. L. Davis, L. C. Hicks, and N. D. Bowers, "The usefulness of time lines in learning chronological relationships in text materials," *J. Experim. Educ.*, vol. 34, no. 3, pp. 22–25, Mar. 1966.

[22] M. Hnida, M. K. Idrissi, and S. Bennani, "Adaptive teaching learning sequence based on instructional design and evolutionary computation," in *Proc. 15th Int. Conf. Inf. Technol. Based Higher Educ. Training (ITHET)*, Sep. 2016, pp. 1–6.

[23] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Inform. J.*, vol. 16, no. 3, pp. 261–273, 2015.

[24] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Berlin, Germany: Springer, 2007, pp. 325–341.

[25] G. Rabby, S. Azad, M. Mahmud, K. Z. Zamli, and M. M. Rahman, "*TeKET*: A tree-based unsupervised keyphrase extraction technique," *Cognit. Comput.*, vol. 12, no. 4, pp. 1–23, 2020.

[26] E. Papagiannopoulou and G. Tsoumakas, "A review of keyphrase extraction," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 2, p. e1339, Mar. 2020.

[27] M. S. U. Miah, J. Sulaiman, S. Azad, K. Z. Zamli, and R. Jose, "Comparison of document similarity algorithms in extracting document keywords from an academic paper," in *Proc. Int. Conf. Softw. Eng. Comput. Syst. 4th Int. Conf. Comput. Sci. Inf. Manage. (ICSECS-ICOCSIM)*, Aug. 2021, pp. 631–636.

[28] T. B. Sarwar and N. M. Noor, "An experimental comparison of unsupervised keyphrase extraction techniques for extracting significant information from scientific research articles," in *Proc. Int. Conf. Softw. Eng. Comput. Syst., 4th Int. Conf. Comput. Sci. Inf. Manage. (ICSECS-ICOCSIM)*, Aug. 2021, pp. 130–135.

[29] B. Jeong, J. Yoon, and J.-M. Lee, "Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis," *Int. J. Inf. Manage.*, vol. 48, pp. 280–290, Oct. 2019.

[30] S. Bag, S. K. Kumar, and M. K. Tiwari, "An efficient recommendation generation using relevant Jaccard similarity," *Inf. Sci.*, vol. 483, pp. 53–64, May 2019.

[31] H. Aljuaid, R. Iftikhar, S. Ahmad, M. Asif, and M. T. Afzal, "Important citation identification using sentiment analysis of in-text citations," *Telematics Informat.*, vol. 56, Jan. 2021, Art. no. 101492.

[32] W. Guo, Q. Zeng, H. Duan, W. Ni, and C. Liu, "Process-extraction-based text similarity measure for emergency response plans," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115301.

[33] R. Singh and S. Singh, "Text similarity measures in news articles by vector space model using NLP," *J. Inst. Engineers (India), B*, vol. 102, no. 2, pp. 329–338, Apr. 2021.

[34] K. Park, J. S. Hong, and W. Kim, "A methodology combining cosine similarity with classifier for text classification," *Appl. Artif. Intell.*, vol. 34, no. 5, pp. 396–411, Apr. 2020.

[35] W. Huang, Z. Wu, P. Mitra, and C. L. Giles, "RefSeer: A citation recommendation system," in *Proc. IEEE/ACM Joint Conf. Digit. Libraries*, Sep. 2014, pp. 371–374.

[36] T. Achakulvisut, D. E. Acuna, T. Ruangrong, and K. Kording, "Science concierge: A fast content-based recommendation system for scientific publications," *PLoS ONE*, vol. 11, no. 7, Jul. 2016, Art. no. e0158423.

[37] P. Winoto, T. Y. Tang, and G. I. McCalla, "Contexts in a paper recommendation system with collaborative filtering," *Int. Rev. Res. Open Distrib. Learn.*, vol. 13, no. 5, pp. 56–75, 2012.

[38] S. Doerfel, R. Jäschke, A. Hotho, and G. Stumme, "Leveraging publication metadata and social data into FolkRank for scientific publication recommendation," in *Proc. 4th ACM RecSys Workshop Recommender Syst. Social Web (RSWeb)*, 2012, pp. 9–16.

[39] H. Raja and M. T. Afzal, "Paper recommendation using citation proximity in bibliographic coupling," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 25, no. 4, pp. 2708–2718, 2017.

[40] O. Kassak, M. Kompan, and M. Bielikova, "User preference modeling by global and individual weights for personalized recommendation," *Acta Polytechn. Hungarica*, vol. 12, no. 8, pp. 27–41, 2015.

[41] D. De Nart and C. Tasso, "A personalized concept-driven recommender system for scientific libraries," *Proc. Comput. Sci.*, vol. 38, pp. 84–91, Jan. 2014.

[42] M. Amami, G. Pasi, F. Stella, and R. Faiz, "An lda-based approach to scientific paper recommendation," in *Proc. Int. Conf. Appl. natural Lang. Inf. Syst.* Cham, Switzerland: Springer, 2016, pp. 200–210.

[43] S. Philip, P. B. Shola, and A. Ovye, "Application of content-based approach in research paper recommendation system for a digital library," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 10, pp. 37–40, 2014.

[44] R. Dong, L. Tokarchuk, and A. Ma, "Digging friendship: Paper recommendation in social network," in *Proc. Netw. Electron. Commerce Res. Conf. (NAEC)*, 2009, pp. 21–28.

[45] K. W. Boyack, H. Small, and R. Klavans, "Improving the accuracy of co-citation clustering using full text," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 64, no. 9, pp. 1759–1767, 2013.

[46] H. Zaugg, R. E. West, I. Tateishi, and D. L. Randall, "Mendeley: Creating communities of scholarly inquiry through research collaboration," *TechTrends*, vol. 55, no. 1, pp. 32–36, 2011.

[47] J. Beel and B. Gipp, "Google Scholar's ranking algorithm: An introductory overview," in *Proc. 12th Int. Conf. Scientometrics Informetrics (ISSI)*, Rio de Janeiro, Brazil, vol. 1, 2009, pp. 230–241.

[48] R. Gove, C. Dunne, B. Shneiderman, J. Klavans, and B. Dorr, "Understanding scientific literature networks: An evaluation of action science explorer," Univ. Maryland, College Park, MD, USA, Tech. Rep., Mar. 2011. [Online]. Available: https://www.umiacs.umd.edu/publications/understanding-scientific-literature-networks-evaluation-action-science-explorer

[49] S. Amjad, H. Mukhtar, and C. Dunne, "Automating scholarly article data collection with action science explorer," in *Proc. Int. Conf. Open Source Syst. Technol.*, Dec. 2014, pp. 160–169.

[50] O. Karnalim, "Language-agnostic source code retrieval using keyword & identifier lexical pattern," *Int. J. Softw. Eng. Comput. Syst.*, vol. 4, no. 1, pp. 29–47, 2018.

[51] K. Barker and N. Cornacchia, "Using noun phrase heads to extract document keyphrases," in *Proc. Conf. Can. Soc. Comput. Stud. Intell.* Berlin, Germany: Springer, 2000, pp. 40–52.

[52] G. Rabby, S. Azad, M. Mahmud, K. Z. Zamli, and M. M. Rahman, "A flexible keyphrase extraction technique for academic literature," *Proc. Comput. Sci.*, vol. 135, pp. 553–563, Jan. 2018.

[53] N. Pradhan, M. Gyanchandani, and R. Wadhvani, "A review on text similarity technique used in IR and its application," *Int. J. Comput. Appl.*, vol. 120, no. 9, pp. 29–34, Jun. 2015.

[54] M. Francis-Landau, G. Durrett, and D. Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," 2016, *arXiv:1604.00734*.

[55] E. Loper and S. Bird, "NLTK: The natural language toolkit," 2002, *arXiv:cs/0205028*.

[56] Python Software Foundation. (2021). *MATH—Mathematical Functions—Python 3.9.1RC1 Documentation*. [Online]. Available: https://docs.python.org/3/library/math.html

[57] Python Software Foundations. (2021). *OS—Miscellaneous Operating System Interfaces—Python 3.9.1RC1 Documentation*. [Online]. Available: https://docs.python.org/3/library/os.html

[58] C. Mattmann. (May 2021). *Chrismattmann/Tika-Python*. Accessed: Jun. 26, 2014. [Online]. Available: https://github.com/chrismattmann/tika-python

[59] F. Boudin, "pke: An open source Python-based keyphrase extraction toolkit," in *Proc. 26th Int. Conf. Comput. Linguistics, Syst. Demonstrations (COLING)*, 2016, pp. 69–73.

[60] G. Rabby. (2020). *Automatic Keyphrase Extraction—Google Drive*. [Online]. Available: http://bit.do/TeKET

[61] *GitHub—TalhaSarwar40/Chronological-Learning-: Article Recommend ation*. [Online]. Available: https://github.com/TalhaSarwar40/Chronological-Learning-

[62] W. Yining, W. Liwei, L. Yuanzhi, H. Di, C. Wei, and L. Tie-Yan, "A theoretical analysis of NDCG ranking measures," in *Proc. Workshop Conf.*, 2013, pp. 1–30.

[63] Z. Zhang and L. Li, "A research paper recommender system based on spreading activation model," in *Proc. 2nd Int. Conf. Inf. Sci. Eng.*, Dec. 2010, pp. 928–931.

[64] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002.

[65] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, "Yake! Collection-independent automatic keyword extractor," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2018, pp. 806–810.

[66] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[67] A. Bougouin, F. Boudin, and B. Daille, "TopicRank: Graph-based topic ranking for keyphrase extraction," in *Proc. Int. Conf. Natural Lang. Process. (IJCNLP)*, 2013, pp. 543–551.

[68] C. Florescu and C. Caragea, "PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Jul. 2017, pp. 1105–1115.

[69] X. Wan and J. Xiao, "Single document keyphrase extraction using neighborhood knowledge," in *Proc. AAAI*, vol. 8, 2008, pp. 855–860.

[70] F. Boudin, "Unsupervised keyphrase extraction with multipartite graphs," 2018, *arXiv:1803.08721*.

[71] M. Ohta, T. Hachiki, and A. Takasu, "Related paper recommendation to support online-browsing of research papers," in *Proc. 4th Int. Conf. Appl. Digit. Inf. Web Technol. (ICADIWT)*, Aug. 2011, pp. 130–136.

[72] Y. Jiang, A. Jia, Y. Feng, and D. Zhao, "Recommending academic papers via users' reading purposes," in *Proc. 6th ACM Conf. Recommender Syst. (RecSys)*, 2012, pp. 241–244.

[73] J. Beel, S. Langer, M. Genzmehr, B. Gipp, C. Breitinger, and A. Nürnberger, "Research paper recommender system evaluation: A quantitative literature survey," in *Proc. Int. Workshop Reproducibility Replication Recommender Syst. Eval. (RepSys)*, 2013, pp. 15–22.

TALHA BIN SARWAR (Member, IEEE) received the Bachelor of Science degree *(magna cum laude)* in computer science and engineering (B.Sc. degree in CSE) and the Master of Science degree (M.Sc.C.S.) in computer science from American International University—Bangladesh (AIUB), Dhaka, Bangladesh. He is currently pursuing the Ph.D. degree with the Faculty of Komputeran, Universiti Malaysia Pahang, Malaysia. He was mentioned in the Dean's List Honor as well. After the consummation of M.Sc.C.S., he joined as a Lecturer at the Department of Computer Science, AIUB. He is a Graduate Research Assistant with the Faculty of Komputeran, Universiti Malaysia Pahang. His research interests include natural language processing, data mining, and machine learning.

NOORHUZAIMI MOHD NOOR received the B.Sc. degree in computer science and the master's degree in science from Universiti Putra Malaysia, Malaysia, in 1999 and 2003, respectively, and the Ph.D. degree in computer sciences from Universiti Kebangsaan Malaysia, Malaysia, in 2016. She has been in the academic, research, and consultancy field, since 2003. She is currently the Head of Program (Entrepreneurship) at the Centre of Creative Entrepreneur Development and a Senior Lecturer at Universiti Malaysia Pahang, Malaysia. She is the author of more than 20 research articles. Her research interests include natural language processing, expert systems, and computer security. She is a Reviewer for the *Journal of Information and Communication Technology* (JICT) and an Editor for the *International Journal of Software Engineering and Computer Systems* (IJSECS). She is also a Certified Professional Technologist from the Malaysia Board of Technologists (MBOT), where she is actively involved as an Assessor Panel for Technology and Technical Academic Programs Accreditation.

M. SAEF ULLAH MIAH (Member, IEEE) received the Bachelor of Science and Master of Science degrees from American International University—Bangladesh (AIUB). He is currently pursuing the Ph.D. degree with Universiti Malaysia Pahang (UMP). He was an Assistant Professor with the Department of Computer Science, AIUB. He is working as a Graduate Research Assistant at the Faculty of Computing, UMP. He is engaged in research and teaching activities and has practical experience in software development and project management. In addition to his professional activities, he is passionate about working on various open-source projects. His main research interests include data and text mining, natural language processing, machine learning, material informatics, and blockchain applications.

MAMUNUR RASHID (Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Pabna University of Science and Technology, Bangladesh, in 2016, and the M.Sc. degree from the Faculty of Electrical and Electronics Engineering Technology, University Malaysia Pahang, Malaysia. He has more than 40 publications. His research interests include brain–computer interface, biomedical signal processing, machine learning, and deep learning. He received the Graduate Research Scheme Scholarship from the University Malaysia Pahang for his M.Sc. studies.

FAHMID AL FARID (Member, IEEE) received the B.S. degree in computer science and engineering from the University of Chittagong, Bangladesh, in 2010, and the M.S. degree from the Faculty of Computer Science and Electrical Engineering, University of Ulsan (UOU), South Korea, in 2015. He is currently pursuing the Doctor of Philosophy (Ph.D.) by research in information technology with Multimedia University, Cyberjaya, Malaysia. From 2013 to 2014, he was a Research Assistant with the Embedded System Laboratory, UOU. In 2015, he was a Research Assistant with the Ubiquitous Computing Technology Research Institute (UTRI), Sungkyunkwan University, South Korea. His current research interests include artificial intelligence, algorithm design, computer vision, human–computer interaction, image and video analysis, power generation, and green technology. He received the Korean BK21 PLUS Scholarship, Supported by Korean Government in M.S. degree, from 2012 to 2014. He also received an ICT Fellowship from Bangladesh Government, in 2014.

MOHD NIZAM HUSEN received the B.Sc. degree in computing from the University of Portsmouth, U.K., in 1997, the master's degree in information technology from Universiti Utara Malaysia, Malaysia, in 2007, and the Ph.D. degree in computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2017. He has been in the academic, research, and consultancy field, since 1997. He is currently a Deputy Dean (academic and technology) and an Associate Professor at the Universiti Kuala Lumpur Malaysian Institute of Information Technology (UniKL MIIT), Malaysia. He is the author of more than 40 research articles. His research interests include intelligent systems, visual attention, pattern recognition, and indoor localization. He is a Reviewer of various well-known journals, among others are the IEEE Communications Letters, IEEE Transactions on Industrial Electronics, *Journal of Pattern Recognition Research*, *Mobile Networks and Applications* (Springer), and *Journal of Location Based Services*. He is also a Certified Professional Technologist from the Malaysia Board of Technologist (MBOT), where he is actively involved as an Evaluation Panel for Technology and Technical Academic Programmes Accreditation. He is also appointed by the government as an Expert Panel for the Ministry of Science, Technology and Innovation (MOSTI) Research and Development Fund for the period of 2021–2023. Other than that, he is also a Certified iOS Application Developer and Certified Android Application Developer.