IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 11, 2021, accepted August 13, 2021, date of publication August 25, 2021, date of current version September 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107939

An Effective Scholarly Search by Combining Inverted Indices and Structured Search With Citation Networks Analysis

SHAH KHALID[1,2], **SHENGLI WU**[1,3], **(Member, IEEE), ABDUL WAHID**[2], **AFTAB ALAM**[4,5], **AND IRFAN ULLAH**[6]

[1]School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
[2]School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad 44000, Pakistan
[3]School of Computing, Ulster University, Belfast BT52 1SA, U.K.
[4]College of Science and Engineering, Hamad Bin Khalifa University, Ar Rayyan, Qatar
[5]Department of Computer Science and Engineering, Kyung Hee University (Global Campus), Yongin-si 17104, South Korea
[6]Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal 18050, Pakistan

Corresponding author: Shah Khalid (shah.khalid@seecs.edu.pk)

ABSTRACT The rapid growth in the number of scholarly documents on the Web and in other digital platforms makes it challenging for researchers to find research publications most relevant to their information needs. This challenge has been mitigated to a greater extent by the major scholarly retrieval systems, such as Google Scholar, Semantic Scholar, PubMed, CiteSeerX, and others. The reason for the success of these retrieval solutions lies in the advances in ranking approaches. However, the existing studies advocate for the fact that we are still far from the method's effectiveness ceiling, leaving ample room for further improvement to meet the scholarly needs of users. The existing methods adopt different approaches; some use classical Information Retrieval (IR), others use semantics-aware methods, including Knowledge Graph (KG) to support scholarly search. However, we hypothesize that combining the best of both worlds can further improve search relevance. In this context, this work incorporates inverted index from the classical IR with BM25 as the weighting scheme, combined with Citation Networks Analysis (CNA) for the baseline search results, which are then re-ranked by passing the selected entities from the top-k initial search results as the search query to the KG. This way, not only the textual content but also the structural semantics of the research publications are well exploited in the retrieval processes. The goal is to exploit IR and KG-based retrieval techniques to gain insights into the behavior of both textual and structured information in the strategic ranking of scholarly articles. The proposed solution has been evaluated using the ACL Anthology Network (AAN) dataset. The results show that the proposed technique can comparatively improve the retrieval performance in terms of Normalized Discounted Cumulative Gain (nDCG) and precision rates.

INDEX TERMS Academic search, knowledge graph, inverted index, structure search, citation networks analysis.

I. INTRODUCTION

The number of publications is growing exponentially. For example, PubMed [1] publishes nearly three thousand articles per day [2]. Searching relevant publications from such a large collection is challenging in terms of time and labor [3]. This is partly due to the large list of research publications that appear in response to the user's search query. For instance, Google Scholar [4] returns 1160K results for the query ''scholarly search systems'' but displays only one thousand papers on its search results page. Nevertheless, it is a daunting task for a researcher to go through all one thousand search results for each query to find relevant publications. To mitigate this issue and reduce the cognitive overload on the user, several prominent research works appeared in recent years in the form of articles [3], [5]–[8], doctoral dissertations [9], test collections [10], books [11], and others. In addition, several academic search engines including Google Scholar,

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

CiteSeerX, Arnetminer, and Microsoft Academic Search assist researchers in finding relevant publications.

The research in the scholarly retrieval domain has employed several methods to improve the relevance of search results. Some of these include query expansion [6], [12], [13], query format and analysis [14]–[17], search results presentation [8], document ranking models [8], [18], [19], exploiting user behavior [20], [21], and recently a multi-objective approach for determining the usefulness of papers [22]. Like other retrieval systems, the ranking of research publications is among the key challenges in scholarly search. However, unlike the classical IR, scholarly retrieval solutions exploit the structure semantics in addition to metadata and content (if accessible, wherein the former textual information is readily available). In addition, scholarly documents have two key parts, namely the textual part and structural part, which hold important informational units including citation networks, co-citation networks, authors networks, etc. These parts, if exploited together, may bring more useful and relevant results. In such a ranking strategy, a citation network may give a lot of information about the quality and the effects of individual paper but catching research topics of the paper as well as how these topics propagate in the search results become challenging. This may worsen the situation in some cases, especially when a large number of documents are retrieved as relevant for a given user query.

While using an academic search engine, researchers examine the top listed publications, retrieved against their search query, in the hope to find the most relevant ones. Sometimes the ranked position matters a little and they go to even the second or third page of the search engine results pages. They may be interested in finding other publications of the same author, their co-authors, and different authors on the same topic. Therefore, ways must be found to consider these aspects in ranking scholarly documents. One possible solution is to merge the search results returned by more than one retrieval engine. This is also highlighted by several authors [23]–[25] that results merging improves search relevance provided that the retrieval solutions used are considered as the independent experts for a particular goal [26]. This article considers this aspect in the retrieval by exploiting Latent Dirichlet Allocation (LDA) topic vectors for topic propagation while traversing citation network for the structural part, and classical IR (for the textual part) of scholarly articles.

This research work exploits a research paper as a combination of textual part consisting of its text (bag of words) and structural part (bag of citations) that comes through the citation network analysis. It is motivated by the idea that results merging from these sources can potentially improve the search results. The inverted index using Apache Solr is combined with the structural index (using Neo4j, a graph database for the KG) for content analysis and exploited together with the citation analysis in the strategic ranking of the scholarly publications. This hybrid approach is tested using retrieval experiments on the Association for Computational Linguistics (ACL) dataset to show its strength against the conventional baselines. The KG construction is inspired by Google, which is constructed by identifying both explicit entities provided by the web pages in the form of semantic labels and implicit entities recognized by contextual analysis through NLP tools [27]. The extracted entities in a KG are enriched for ideal usability through ontologies and knowledge base data. Likewise, for scholarly KG, the same approach can be applied [28]. Just as web pages contain multiple forms of data, scholarly documents also contain multiple types of components, i.e., title, authors, abstract, full-body text, tables, algorithms, figures, and citations. All these can be extracted along with their relationships and metadata for enriching KG [27]–[33].

The mechanism of structural index in Neo4j follows the same method for KG bootstrapping from scholarly documents as used in [29], which follows the same definition and procedure for building KG as practiced in DBpedia [34], WordNet [35] and Probase [36]. However, it focuses only on a hyponym relation among the entities to explore the possibility for scientific entity-based search [29]. In comparison, the proposed approach builds an academic KG using the ACL dataset without using any external source. The KG includes concept entities, their descriptions, context correlations, and relationships (citations) with authors and co-authors. It stores the citation network and integrates the citation information with the paper's content to discover topic propagation on the citation network. We apply this KG to our hybrid ranking task to search graph structure by using LDA for topic distribution.

Recently, Semantic Scholar introduced explicit semantic ranking for scholarly search [8], which connects both queries and documents using semantic information from KG. The KG holds concept entities, their descriptions, and relationships with authors, venues, and embeddings trained from the graph structure. The constructed KG and embedding are used for explicit semantic ranking [8] using the Semantic Scholar (S2) corpus, query log, and freebase [8].

The proposed framework searches the indexed research paper collection using Apache Solr, where BM25 is applied in weighting documents against the search query and merges the results with the structural search techniques using citation network analysis. What makes the proposed technique different, is its exploitation of both the inverted indices and structure search (IS) capability of Neo4j by propagating citation network analysis in the strategic ranking of scholarly documents; therefore, we called it Inverted Indices and Structure Search with Citation Analysis (ISCA). The novelty of ISCA lies in the combined exploitation of graph data management techniques and the IR language model with citation analysis that may bring the most relevant papers for a given search query in a more nuanced way, effectively and efficiently. The following are the key contributions of this work:

- A new research problem of combining classical IR and structured search is studied to improve search relevance.
- An effective inverted index, KG, and citation networks are designed and maintained from ACL papers collection.

- ISCA: a hybrid technique by combining an inverted index, structured search KG, and citation networks analysis is proposed for supporting the scholarly search.
- The effectiveness of ISCA via standard evaluation metrics is presented to understand the positive impact of combining the textual and structural information on the relevance of search results.

The rest of the paper is ordered as follows: Section II presents related work, Section III presents an overview of the framework. Section IV describes the experimental setup and performance comparison. Section V concludes the paper with the identification of some future research directions.

II. RELATED WORK

In the context of this paper, the current related works can be categorized as scholarly search and ranking methods, which are briefly summarized in the following subsections.

A. SCHOLARLY SEARCH

The rapid growth of the scholarly documents and their resulting information overload has motivated the development of academic search systems including Google Scholar, Semantic Scholar, AMiner, Xueshu Baidu, Microsoft Academic search, CiteSeerX, and many more. Among these, Google Scholar has been dominant concerning its coverage to the literature [37]. However, greater coverage is not always sufficient, especially from the perspective of scholars seeking relevant documents, where ranking plays an important role. The problem is, Google Scholar and similar other solutions rely heavily on the citation count to rank papers among the list of related candidates. Therefore, if ranking remains unsuccessful, the scholars overpass coverage and go to other scholarly platforms including CiteSeerX [38], AMiner [39], PubMed [40], Microsoft Academic Search [41], and Semantic Scholar [8]. These search engines use analytical tasks including author name disambiguation [39], paper importance modeling [42], and entity-based distinctive summarization [43].

The most prominent research question regarding the academic search is how to define and measure that two articles are similar and related concerning a given search query. Considering the exploitation of textual and structural information, as this research suggests, the most relevant work is by Mai *et al.* [25], which used textual and structural embeddings to find the relevant papers against the given search query. However, it requires training data and parameter settings in scholarly retrieval [25] in a supervised manner, which is different from our unsupervised approach.

B. THE RANKING METHODS

This section discusses the widely used ranking methods used in the academic search domain that are closely related to the proposed work.

1) KNOWLEDGE-GRAPH-BASED METHODS

Several studies have exploited KG in ranking documents against the search query. Explicit semantic ranking uses KG embedding in ranking scholarly documents [8] by experimenting with Semantic Scholar corpus, query log, and freebase to build an academic KG. Its KG considers concept entities and their descriptions, context correlations, relationships with authors and venues, and embedding trained with the graph structure. It employs learning to rank on the search query and represents documents (as entities in the KG) in the embedding space.

Al-Zaidy and Giles [29] constructed KG by extracting entities having only the hypernym-hyponym relationships from a set of ten thousand articles. Such a relationship exists if one entity is an instance of the concept represented by the other entity. We adopt the same method for KG construction while harvesting about twenty three thousand articles available in the AAN dataset [44] and exploit CNA as well. Intuitively, the use of both the key functions (BM25 for textual similarity) and KG (for structural search)) with CNA becomes the multiplier for a change in the strategic ranking of scholarly articles. For extracting semantic relations for scholarly KG construction, Al-Zaidy and Giles [45] extracted entities as concepts and instances along with their attributes from the entire content of the scholarly documents. They used an iterative algorithm for extracting two types of relationships, i.e., concept-instance relationship, and property relationships for taxonomy construction without using any external source such as DBpedia, Wikipedia [46] or WordNet [35].

Unlike the traditional web search, a large portion of queries received by the scholarly retrieval systems is related to entities. It is estimated that more than 50% and 70% queries received from Semantic Scholar and Bing, respectively, are related to entities [8], [47]. Such entities not only reflect user needs but also help model query document relevance using the bag-of-entities representation [48]. We hypothesize that if entities are extracted from the top-k results returned by the initial query and run over the KG, a refined list of papers with improved relevance could be obtained. We aim to achieve this with the proposed technique, ISCA, which combines the best of the inverted and structural KG indices with CNA to support the academic search experience of research scholars.

2) CONTENT-BASED METHODS

The content-based approaches rank related articles for a given user query [7] by processing the papers' textual content including title, abstract, keywords, and body. These methods weight relevant articles according to the frequency and position of terms in the article. Several techniques use the term weights to estimate the relevance of articles. Among these, the most widely used approach is the vector space model, which represents each article as a vector of term weights and measures relatedness using cosine similarity. It is practiced by several current retrieval platforms such as Apache Lucene, Solr, ElasticSearch, etc., although cosine similarity does not perform well in many situations [48], [49].

The vector representation of the vector space model was improved further by Latent Semantic Analysis (LSA) using singular value decomposition [50], [51]. However, LSA is unable to perform well for scholarly articles either [49]. To further improve the ranking performance, several other retrieval techniques and weighting schemes including BM25, PubMed, etc., were introduced. The Best Matching 25 (BM25) [52] is a probabilistic IR approach to weigh scholarly documents against the search query [48]. PubMed is a popular scholarly retrieval system primarily designed for biomedical literature [7]. It considers many factors of an article for indexing and retrieval of documents, including (a) number of terms in the article Term Frequency (TF); (b) position of terms (i.e., title, abstract, body-content); (c) weight of terms in the article; and (d) key terms of the article in a domain-specific database, e.g., Medical Subject Headings.

Many factors affect the effectiveness of scholarly retrieval systems. These include collection features, the methods of ranking, the algorithm complexity of exploiting citation networks, the exponential growth of the literature, etc. In recent work, we used both query expansion and CNA for supporting the scholarly search. However, none of the current approaches exploited together the classical IR (BM25) and KG with CNA to support the academic search. Many factors influence the effectiveness of scholarly retrieval systems. These include collection features, ranking methods, the complexity of algorithms for exploiting citation networks, the exponential growth of the literature, etc., [7], [15]. In a recent work [6], we used both query expansion and CNA for supporting scholarly search. However, none of the current approaches jointly used the classic IR (BM25) and KG with CNA to support the academic search.

III. PROPOSED FRAMEWORK

The abstract architecture of the proposed ISCA is shown in Figure 1. It has three layers, including Scholarly Data and Knowledge Curation (SDKC), Processing and Retrieval Business Logic (PRBL), and User Interface (UI). The SDKC layer pre-processes, manages, and indexes the scholarly data. The ACL collection is pre-processed to extract the entities and relations in generating inverted and structure indices as well as citation networks. The PRBL layer is responsible for the retrieval having two components, namely scholarly data retriever and query & results processor. The scholarly data retriever uses BM25, citation analyzer, metadata analyzer, and structure searchability of Neo4j (vector space model). Additionally, the query & results processor processes the search query, extracts entities, and integrates both the initial and final results. The UI layer allows users to interact with ISCA. It holds four components including scholarly graph visualizer, query and results formatter, request, and response handler. The UI layer supports the creation and refinement of the user queries and displays results in both graph and textual formats.

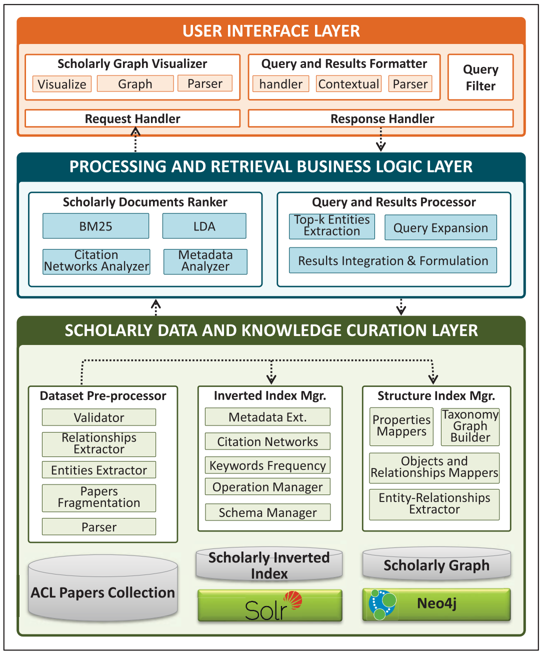


FIGURE 1. The proposed ISCA architecture.

The proposed ISCA supports full-text search by using inverted index and CNA on the one hand, and a KG on the other hand to maintain a graph representation of the same corpus. The inverted index and citation network use classical IR for retrieval against a given search query. The top-*k* results of the initial search results are processed by ISCA to extract interesting entities (paper ID, title, year, authors, and topics distribution of papers) and run it over the KG. Moreover, topic vectors by employing the LDA algorithm and store in KG to visualize topic propagation in the citation network. In this way, the KG refines the initial results by navigating a structured repository, selecting additional relevant documents (using full-text search and clustering capability of Neo4j), or reinforcing the relevance of baseline results produced by the inverted index via the IR model and citation analysis. Figure 2 schematically illustrates the flow of ISCA.

A. SCHOLARLY DATA AND KNOWLEDGE CURATION LAYER

The `SDKC` layer transforms, pre-processes, and standardizes the scholarly dataset for creating inverted and structure indices. This layer comprises three components, i.e., Dataset Pre-processor, Inverted Index Manager, and Structure Index Manager. The Dataset Pre-processor parses the datasets, responsible for extracting text, metadata, and relationships. It refines the extracted data through a metadata analyzer. This way, this step extracts the significant fragments including title, authors, abstract, venue, citations (incites and out-cites), metadata, and entities from research articles by employing the `paper fragmenter` and `entities extractor` modules, respectively. The entities are then harvested and their relationships from the set of parsed sentences are extracted using title and abstract. For entity recognition,

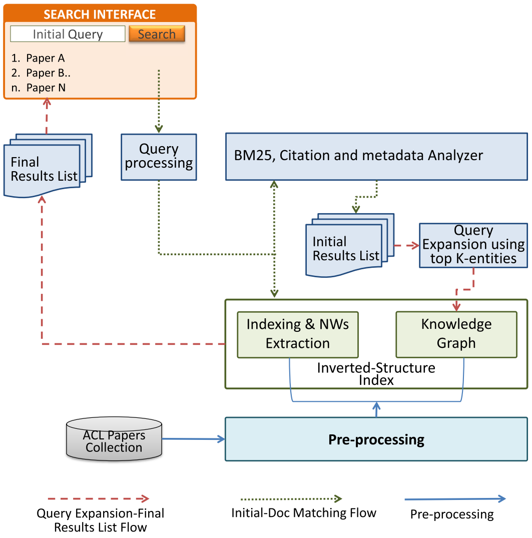


FIGURE 2. The workflow diagram of ISCA.

the nouns and noun-phrases are extracted by following [45]. The `relationships extractor` establishes the relationships among entities for KG construction. For KG construction, two types of relationships, namely hyponym and entity properties, are exploited. The hyponym edge exists if one entity is an instance of the concept represented by the other entity. This is also called `is-a` relationship and can exist between concept and its sub-concepts as well as concept and its instances. The properties relationship is called an `isPropertyOf` relationship, which exists between entities and attributes. For example, from the phrase ''efficient search engine'', we extract 'efficient' as an attribute of the search engine.

The proposed approach borrows notations from [45]. For a paper P_i, a set of sentences S_i having candidate tuples (A_i, B_j) are extracted such that, $S = (A, B_1), (A, B_2) \ldots (A, B_n)$, where A is a hypernym for candidates hyponyms B_j. The extraction gives the pairs $P = (A_1, B_1), (A_2, B_2) \ldots (A_n, B_n)$ for KG construction. For the key entities and generalizability, ISCA also plots interesting relationships for various purposes, including e.g., co-author, authors connected to other papers on the same topic, etc.), as schema independently delineated in Figure 3. Articles are clustered using the clustering capability of KG base on authors and topics.

`Inverted Index Manager` produces the inverted index with pairwise postings consisting of paper ID, title, abstract, main-content, and citations (incites, out-cites). The following two aspects are considered in the strategic ranking of scholarly documents.

1) TOPIC PROPAGATION AND CITATION NETWORKS

A citation network is a directed graph with nodes representing the papers and links denoting citation relationships among them. According to Egghe and Rousseau [53], the citation in paper P_i for paper P_j is represented by an arrow from node representing P_i to P_j. In this way, the papers from a collection

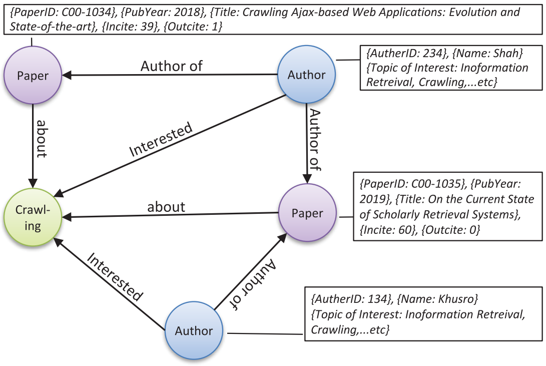

{PaperID: C00-1034}, {PubYear: 2018}, {Title: Crawling Ajax-based Web Applications: Evolution and State-of-the-art}, {Incite: 39}, {Outcite: 1}
Paper
Author of
Author
{AutherID: 234}, {Name: Shah} {Topic of Interest: Inoformation Retreival, Crawling,...etc}
about
Interested
Author of
{PaperID: C00-1035}, {PubYear: 2019}, {Title: On the Current State of Scholarly Retrieval Systems}, {Incite: 60}, {Outcite: 0}
Crawl-ing
about
Paper
Interested
Author of
Author
{AutherID: 134}, {Name: Khusro} {Topic of Interest: Inoformation Retreival, Crawling,...etc}

FIGURE 3. General relationships view.

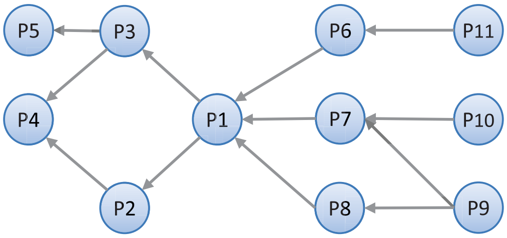


FIGURE 4. The *CitingPapersSet* and *CitedPapersSet*.

C form a directed graph, called a ''citation graph'' or ''citation network.'' Given a citation network, as shown in Figure 4, which is a directed graph $G(P, E)$ consisting of nodes and edges. In $G(P, E)$, P represent the set of nodes, i.e., research papers, where E is a set of directed links showing citing relationships among papers. For each paper (node) n of P, we delineate the notion of the following two sets, that we use during citation analysis in topic propagation:

- $CitingPapersSet(n) = \{m \in P \mid$ there is a link from n to $m\}$. The $CitingPapersSet(n)$ represents research papers cited by paper n directly or indirectly. They are articles influencing paper n.
- $CitedPapersSet(n) = \{m \in P \mid$ there is a path from m to $n\}$. The $CitedPapersSet(n)$ represents the set of publications that either cite article n or influences it.

Figure 4 shows the concept of *CitingPapersSet* and *CitedPapersSet* used in this research to implement functions regarding topic propagation in KG. It can be seen that $CitingPaperSet(P_1) = \{P_2, P_3, P_4, P_5\}$ and $CitedPaperSet(P_1) = \{P_6, P_7, P_8\}$. In Citation Network (CN), each node holds the basic properties of research paper such as paper ID, title, publication year, and authors. This work introduces one additional property named ''Topic Vector (TV),'' gained by the LDA algorithm for topic distribution of a research paper, discussed in the next section.

In the graph database of the citation network, we use two text files created from the ACL corpus. 1) papers.txt 2) citation.txt; the former holds the information about each node includes paper id, year, title, and TV (gained from the LDA model), the latter contains the information about the connection (link) between nodes, i.e., the start and end nodes of each edge as follows:

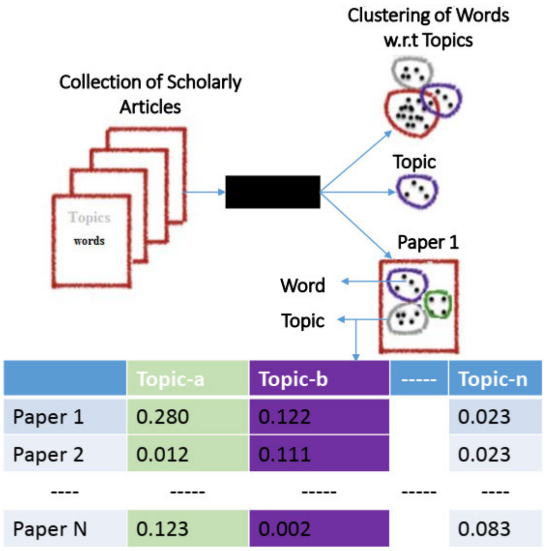
```
PaperID   Year  Title                 TopicVector
P14-5010  2014  The standford core....  0.15..
//PaperID    # Link    PaperID
P14-5010     # Ref     P15-5401
P15-7233     # Ref     P21-7621
P12-2336     # Ref     P11-3740
P10-1723     # Ref     P01-1126
..........
```

While using Neo4j, two main entities namely node and edge are used. The nodes and properties are created, as described in papers.txt and citation.txt, we use createNode() and setProperty(), respectively. Likewise, createRelationshipTo() and setProperty() are used to create edges and set the properties of the edges.

2) DISCOVERING TOPICS WITH LDA

To discover latent topics in papers for topic propagation, this study uses LDA, which considers each paper as a bag of words. For illustration, let $Dic = W_1, W_2, W_3, \ldots W_V$ is dictionary of a corpus C with V words and $C = P_1, P_2, P_3, \ldots P_N$ is the set of papers with N documents (papers), where paper $P_i = W_{i1}, W_{i2}, W_{i3}, \ldots W_{iq}, W_{ij} \epsilon Dic$ and $q \leq V$. With the above corpus 'C' and integer 'L' as input, LDA treats a document as a distribution of L topics and each topic is a distribution of V words using the Gibb Sampling algorithm [54]. Given $T = t_1, t_2, t_3, t_4, \ldots t_L$ as a set of L topics to be discovered from the collection C of papers P, when: $t_i = w_{p1}, w_{p2}, w_{p3}, \ldots w_{pV}$ with $w_{pi} \epsilon [0, 1]$ and $\sum_v^1 wp_i = 1$ is the distribution of words of topic i in the collection. Using this topics' definition, LDA finds latent topics in a research paper and presents each one as: $TV_i = tp_{i1}, tp_{i2}, \ldots tp_{ij}$, with $t_{pi} \epsilon [0, 1], j \leq L$ and $\sum_v^1 tp_i = 1$. This represents the distribution of L topics for document i in the collection C. Therefore, LDA is exploited as a classification method that is both multi-class and multi-label.

In this research, TV_i is the topic vector of the corresponding paper. Each node in the network has a topic vector discovered by the LDA model. This property of a given node is used in structure search. We use a threshold to identify whether a specific paper contains this topic or not. For example, Figure 5 schematically describes how to determine this threshold and how it works for topic distribution with respect to the collection of papers. As we see in Figure 5, paper 1 comprises of three topics, let it has 20% topic a (green), 42% topic b (purple), and 38% topic c (gray). The topics that represent a given paper can be found either by picking their top K probabilities or by setting a threshold for probability and picking only those topics that have probabilities higher than or equal to the threshold value. For example, we can see that paper 1 mainly focuses on topic a (green) and topic b (purple) and less on topic c (gray). With tuning and several experiments on the given ACL collection, as it contains about 23058 documents, we found 0.15 as an optimal threshold in our settings. Researchers will easily capture general sight about individual papers and their citing and cited articles with this visual information. The main objective of integrating the concept of LDA via citation network in KG is to propagate the topic distribution, which may bring the most relevant papers in the search results list against the given search query. For this, we develop some graph query functions to analyze topic propagation through the network during structure search.



	Topic-a	Topic-b	-----	Topic-n
Paper 1	0.280	0.122		0.023
Paper 2	0.012	0.111		0.023
----	-----	-----	-----	----
Paper N	0.123	0.002		0.083

FIGURE 5. LDA topic distribution and modeling.

The prerequisite and important step for entities-based ranking via KG embedding is the KG construction that can store information and relationship between the entities. The `structure index manager` generates the graph by harvesting the ACL dataset without using any external source. The concept entities are extracted from the scholarly documents by following [29], [45] and connected through edges by following [45].

B. PROCESSING AND RETRIEVAL BUSINESS LOGIC LAYER

The core business logic of ISCA holds two components, namely document ranking, and query & results processor. The document ranking is the key component that assimilates the ranking results from the inverted index with the results from the KG-based search. Since ranking techniques in graph databases using keyword-based queries have been extensively studied in the domain of databases [55], the scenario here is different, as keyword-based search over graph databases consider the occurrences of query terms in the document and returns a ranked list of non-redundant Steiner trees or subgraphs [56]–[58]. Instead of being restricted to classical retrieval methods [59], we leverage the graph structure for searching and ranking by aggregating the weights of nodes and edges, attribute-value statistics with content-based relevance measures using the full-text searching capability of Neo4j using Vector Space Model (VSM). Two algorithms are presented here to describe the flow systematically.

Algorithm 1 is the leading ISCA retrieval algorithm, which calls Algorithm 2 in step-1 for retrieving the initial results list using BM25 and CNA [6].

Algorithm 1 ISCA Retrieval

```
Input  : $Q \leftarrow query, P \leftarrow paper$
Output: $list\ of\ relevant\ papers$
1 $InitialRetr \leftarrow Call(Algorithm - 2(Q, P))$
2 $E_{Top_k} \leftarrow Top\ K\ results\ in\ InitialRetr$
3 $KG \leftarrow E_{Top_k}$ /* Pass to KG */
4 $KG_{Retr} \leftarrow (score(E_{Top_k}, \mathrm{KG}))$
5 $F_R\ =\ using\ Equation\ 3$
```

Algorithm 2 BM25 and CA[35]

```
Input  : $Q \leftarrow query, P \leftarrow paper$
Output: $list\ of\ papers$
foreach $paper\ P_i\ in\ N$ do
    $Compute\ base\ weight\ using\ BM25$
end
$P_{IR} \leftarrow W_i$
while $W_i \neq 0$ do
    foreach $W_i$ do
        $Compute\ each\ paper\ citation\ score$
        $using\ citation\ networks\ analysis$
    end
end
return $P_{IR}$
```

Algorithm 1 extracts entities from the top-k results including TVs of the initial results set in step-2 and passes it to the KG in step-3. Step-4 uses citation network analysis in amalgamation with LDA for topic propagation and connected entities to the same concept with structure clustering statistics for generating and presenting the KG-based retrieval. The integrated final results list in step-5 is presented using Equation 3.

To understand the topic propagation from top k papers in initial search results, we need to get the corresponding *CitingPapersSet* and *CitedPaperSet* (as defined in Section 3.1) of these top papers. The LDA [1] method then discovers latent topics in these papers that are collectively related to the interesting topic. Algorithm 3 describes this procedure which follows and adopts the approach presented in [23]. The related papers found by Algorithm 1 are then used to get citing/cited set along with the publication year for including the freshness factor from the graph database and then refine the final results list.

The ranking of the ISCA is divided into two main steps: a) initial ranking b) final ranking. The initial ranking generates the baseline results list against the search query of the users using an inverted index via BM25 and CNA, as shown in Figure 6. The approach uses both BM25 and CNA as citations play a vital role in the evaluation of scholarly papers [6], [60].

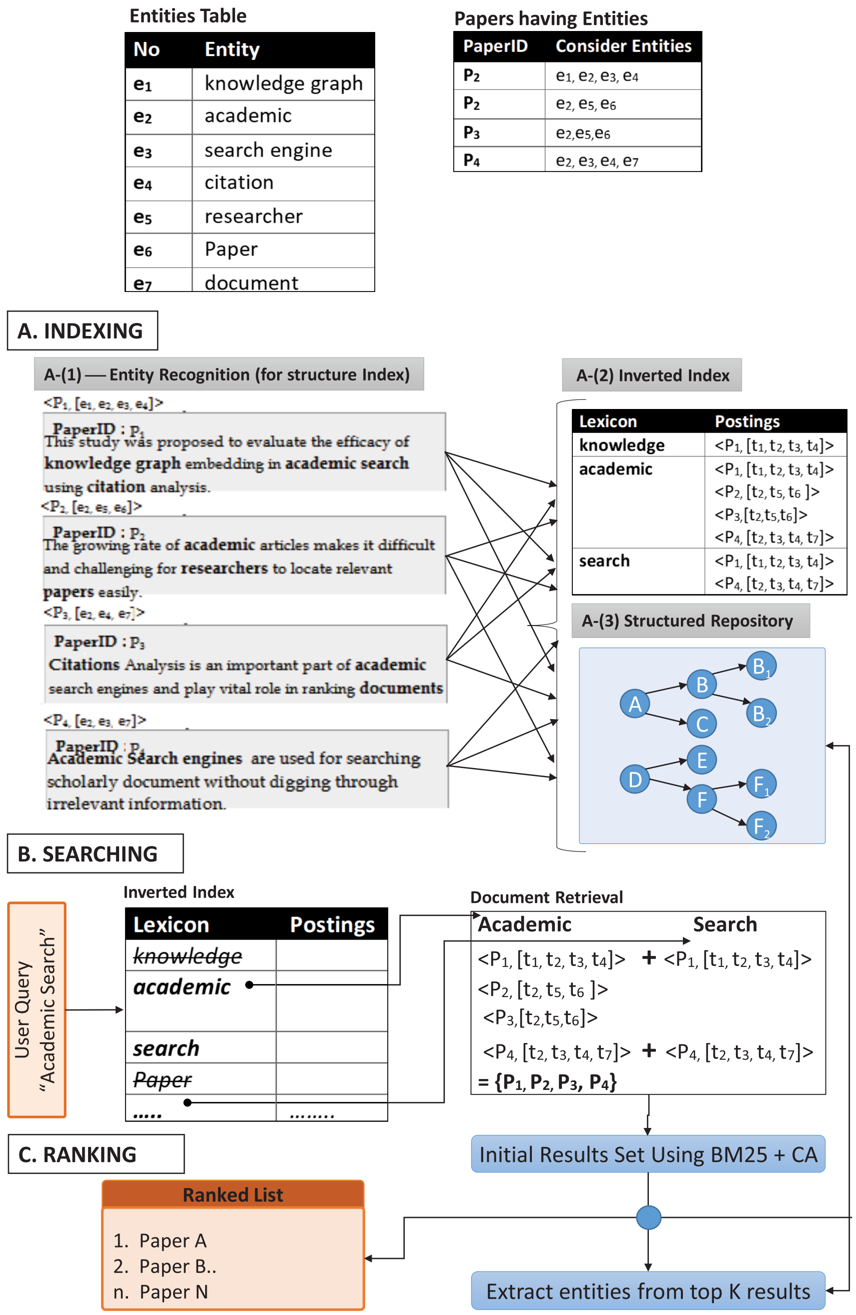
Algorithm 3 Topic Propagation Using Cited and Citing Papers Sets

```
Input  : $PaperId$ of the root papers(nodes), $TopiId$ to
         evaluate, $Integer$ indicates direction ($-1$ shows
         $CitingPaperSet$ of $PaperId$, 1 shows
         $CitedPapersSet$ of $PaperId$)
Output: Papers in $CitedPapersSet/CitingPapersSet$
         related to topic $TopicId$
$resultsList \leftarrow null$
$R \leftarrow null$
if $direction = -1$ then
    /* Described in Section III(A)-1 */
    $R \leftarrow CitingSet(PaperId)$
else
    /* Described in Section III(A)-1 */
    $R \leftarrow CitedSet(PaperId)$
end
foreach $Paper\ P\ in\ R$ do
    if $P\ related\ to\ topic's\ TopicId$ then
        Add $P$ to the results list
    end
end
return $P_{IR}$
```

For the citation network analysis, it considers the output of the CNA, extracts, builds, and uses citation networks to compute citation scores besides textual similarity. To calculate citation score, we use the PageRank [61] algorithm on the citation network with the assumption that it can be viewed as an "up-vote" from one article to another, i.e., the article having higher "up-votes" can be considered the most important and influential besides their textual similarity. For each connected paper P_i, we use Equation 1 to compute the citation score, and through this, we can find the most influential articles within the given citation graph.

$$PR_{P_i} = \frac{(1-d)}{N} + d\sum\nolimits_{P_j \in in(P_i)} PR_{P_j}.S_{in(P_i,P_j)}.S_{out(P_i,P_j)} \tag{1}$$

In Equation 1, the constant d is set to 0.85, N represents the number of publications in the citation network and ($inPi$) is the set of all the nodes which are in-links to P_i. $S_{in}(P_i, P_j)$ is the score of the link (P_i, P_j) estimated based on the number of incites of P_i and the number of incites of all reference papers of P_j. $S_{out}(P_i, P_j)$ is the weight of link (P_i, P_j) estimated based on the number of out-links of P_j and the number of out-links of all reference papers of P_i. The results of the citation network analyzer are stored as node properties in KG. To amalgamate the normalized CNA score in the ranking, two types of influence marks are used for each edge of the paper, i.e., strong influence mark (SI+) and weak influence mark (WI-) for each connected paper, which also helps eliminate



Entities Table

No	Entity
e_1	knowledge graph
e_2	academic
e_3	search engine
e_4	citation
e_5	researcher
e_6	Paper
e_7	document

Papers having Entities

PaperID	Consider Entities
P_2	e_1, e_2, e_3, e_4
P_2	e_2, e_5, e_6
P_3	e_2, e_5, e_6
P_4	e_2, e_3, e_4, e_7

A. INDEXING
A-(1) — Entity Recognition (for structure Index)
<P1, [e1, e2, e3, e4]>
PaperID : P1
This study was proposed to evaluate the efficacy of knowledge graph embedding in academic search using citation analysis.
<P2, [e2, e5, e6]>
PaperID : P2
The growing rate of academic articles makes it difficult and challenging for researchers to locate relevant papers easily.
<P3, [e2, e4, e7]>
PaperID : P3
Citations Analysis is an important part of academic search engines and play vital role in ranking documents
<P4, [e2, e3, e7]>
PaperID : P4
Academic Search engines are used for searching scholarly document without digging through irrelevant information.
A-(2) Inverted Index
Lexicon
Postings
knowledge
<P1, [t1, t2, t3, t4]>
academic
<P1, [t1, t2, t3, t4]>
<P2, [t2, t5, t6]>
<P3,[t2,t5,t6]>
<P4, [t2, t3, t4, t7]>
search
<P1, [t1, t2, t3, t4]>
<P4, [t2, t3, t4, t7]>
A-(3) Structured Repository
A
B
B1
B2
C
D
E
F
F1
F2
B. SEARCHING
User Query
"Academic Search"
Inverted Index
Lexicon
Postings
~~knowledge~~
academic
search
~~Paper~~
.....
....
Document Retrieval
Academic
Search
<P1, [t1, t2, t3, t4]> + <P1, [t1, t2, t3, t4]>
<P2, [t2, t5, t6]>
<P3,[t2,t5,t6]>
<P4, [t2, t3, t4, t7]> + <P4, [t2, t3, t4, t7]>
= {P1, P2, P3, P4}
Initial Results Set Using BM25 + CA
C. RANKING
Ranked List
1. Paper A
2. Paper B..
n. Paper N
Extract entities from top K results

FIGURE 6. Schematic flow of indexing, searching, and ranking.

the non-relevant connected papers as given in Equation 2.

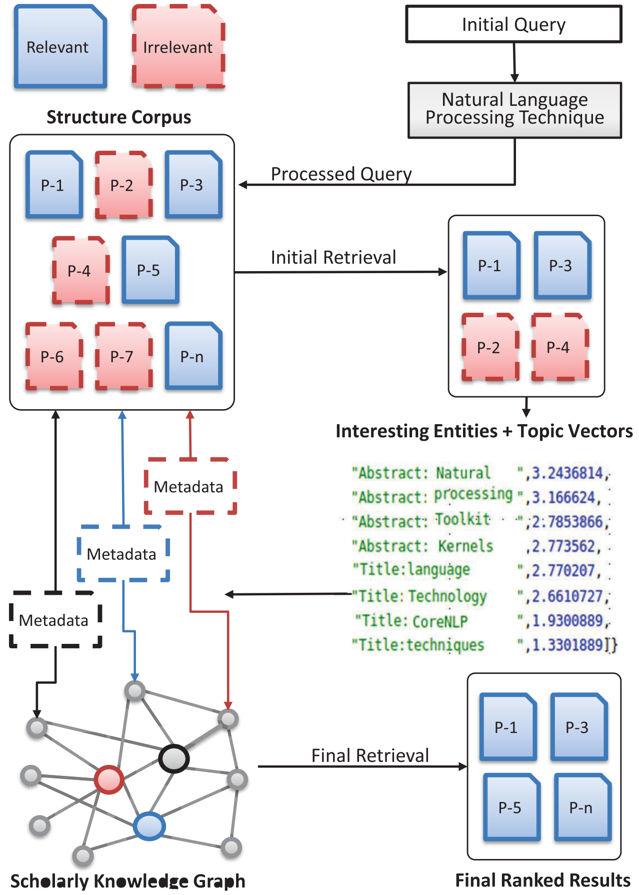
$$
if \begin{cases} S(P_i) = 1, e_{ij} = SI+ \\ S(P_i) = 0, e_{ij} = WI- \end{cases} \tag{2}
$$

For instance, when P_i connects to P_j, if P_j holds query term (directly or indirectly) in the specified fields using similarity function $S(P_i) = Similarity(q, (P_i))$, then we give a strong influence mark to the edge e_{ij} (i.e., 1), and weak influence mark (i.e., 0), otherwise. For KG-based retrieval, the approach automatically generates an entity-based query from the top k results of the initial results list and passes it to the KG. It also applies a threshold τ on the (q, e, p) to filter the entities that do not match well so that their overhead can be minimized. In other words, to include only entities from those papers in the expanded queries for which $(q, e, p) > \tau$. These extracted entities are used as seeds in the graph. From these seeds (nodes), it traverses the graph following the promising edges that may lead to more relevant additional results by employing the clustering plus full-text searching capability of Neo4j. The sample query for searching the top 20 most influential articles posing to structured repository looks as follows:

```
MATCH (p: Paper)
RETURN a.title as paper,
a.pagerank as score
ORDER BY score DESC
LIMIT 20
Results:
```

Likewise, if we want to find an article for the expanded entities-based query formulated from the extracted entities in Figure 7, i.e., "Natural Language processing toolkit," the sample query looks as follows:

```
$MATCH (e:Entity)-[:DESCRIBES]->()
<-[:HAS_ANNOTATED_ENTITY]-(p:Paper)$
$WHERE e.value = "Natural  Language"
OR/AND "Processing Toolkit".$
$RETURN p.title as title,
p.pagerank as p$
$ORDER BY p DESC$
$LIMIT 20$
```



```
"Abstract: Natural    ",3.2436814,
"Abstract: processing",3.166624,
"Abstract: Toolkit    ",2.7853866,
"Abstract: Kernels    ,2.773562,
"Title:language       ",2.770207,
"Title: Technology    ",2.6610727,
"Title: CoreNLP       ",1.9300889,
"Title:techniques     ",1.3301889]}
```

FIGURE 7. A general retrieval scenario using KG.

The final results are produced by merging initial results with those of KG-based retrieval using Algorithms 1 and Algorithm 2. This is obtained by merging the initial results list from the IR model and CNA (using step 2 to 10 of Algorithm 2) and KG-based retrieval (using step 2 to 5, Algorithm 1), as described in Equation 2.

Let $InitialRetr = P_1, P_2 \ldots P_n$, obtained from the inverted index for user keywords query, and $KGRetr = P_1^\circ, P_2^\circ, \ldots P_n^\circ$ obtained from the KG by transitively exploiting entities-based query. Once a new results list of papers reaches from the KG, there is a need to devise a strategy that includes ranking them by employing a scoring function and merging *InitialRetr* with the *KGRetr* to get the final search results. For instance, given a paper $P^\circ \in KGRetr$ and $P \in InitialRetr$ from which P° generated using KG. For P°, we compute the ranking score using the full-text search capability of Neo4j's (VSM model) in amalgamation with graph clustering ability using CN to locate relevant connected papers through the same entities (concepts) by scoring (q, p, p°). This scoring function exploits the influence of connected entities through the properties attached and CNA above the textual similarity.

The merging strategy aims to include as many relevant papers as possible at the top of the final results set, i.e., to give related papers higher ranks during merging. For instance, if a result set has many relevant documents in the top ranks, then these top-ranked relevant articles should be included in the final results set. Also, if a result set has few or even no relevant papers, then the final results list should have a minimum of them. Furthermore, as we have two retrieval systems, i.e., IR-based and KG-based retrieval systems, the results list generated by them may have different score ranges, which need normalization before merging. This normalization is achieved by following the relevant literature [26], [62]–[64]. Let $L_{IR,q}$, and $L_{KG,q}$, respectively represent the results lists produced by these retrieval systems for the query q. Here, $L_{KG,q}$ is the transitive results list of the entities-based query expanded from q. A fusion strategy is used to merge these lists to produce a unique ranked list of papers for the given query q. Let S_{IR} is the IR model score, and S_{KG} is the KG score computed by the dependent IR model. Then a linear model described in Equation 3 is used to obtain the final score.

$$S_{p_i,q} = \alpha . S_{IR} * (1-\alpha) \sum\nolimits_{(e_i=1)}^{n} e_i * S_{KG} \tag{3}$$

where q is the user keywords query, S_{IR} and S_{KG} are the normalized scores, $\alpha[0, 1]$ is a weighting coefficient, and e_i is the entity score of paper P_i. Equation 4 normalizes the scores for merging.

$$S_n = (S - S_{Low})/(S_{High} - S_{Low}) \tag{4}$$

In Equation 4, S_n is the normalized score, where S_{Low} and S_{High} are the lowest and highest limits in each run. For more explication, the following section presents the working flow of the proposed approach.

C. UNDERSTANDING THE FRAMEWORK USING AN EXAMPLE-QUERY

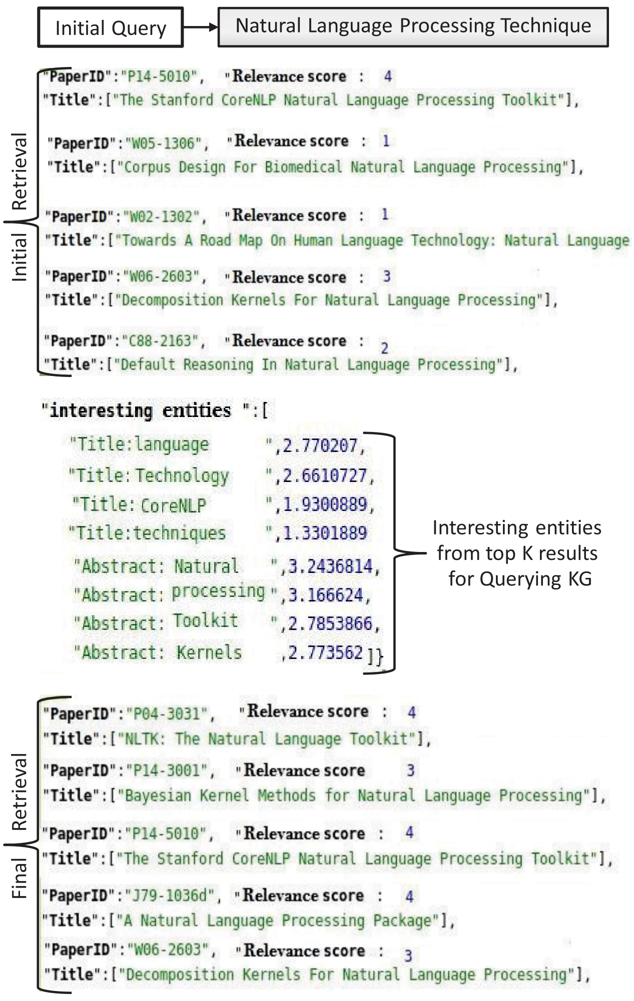
Figure 6, schema independently describes how the ISCA performs indexing, searching, and ranking. It demonstrates the entire procedure in three key steps. Step ''A'' presents the creation of the inverted and structured repositories. The indexing subsystem of the framework for every paper *P* extracts entities and makes paired posting, i.e., *PaperId* and a list of entities for the creation of a structured repository (KG). This step also creates an inverted index accordingly. After indexing, it presents the process of searching for user keywords query in step ''B''. For query expansion and formulation, several other mechanisms like pseudo-relevance techniques are employed on top of BM25 to improve and enrich the ranking mechanism for a detailed discussion about query reformulation and entity-based search in KG visit [65], [66]. In step ''B'', the user enters keywords query regarding a specific research topic, the system processes it against the inverted index using IR model and transitively passes the extracted entities as an entity-set query to KG.

Step ''C'' pictorially presents the initial and final ranking using Equation 3, i.e., the way how the framework uses the IR language model along with CNA and KG to enhance the final retrieval. To highlight the effect of KG in the ranking, we describe a general retrieval scenario and example-query, as shown in Figure 7, and Figure 8 respectively.

1) GENERAL RETRIEVAL SCENARIO

Figure 8 presents example-query to substantiate how the proposed approach refines the final results set, i.e., the results before and after querying the KG. The average user relevance judgments show that this hybrid technique can bring more relevant papers to the top of the search results list. The schema-independent results with user relevance to the academic searcher query are explicated in Figure 8, which presents the effectiveness of KG in retrieving relevant papers.

In KG construction, we keep care of two characteristics: a) entities are interlinked, we hypothesize that related entities are often connected and may be used effectively during graph traversing. b) The same concepts are clustered in the graph, such that many entities can be queried iteratively following the links from the same entity in the same cluster. Here, the clustering ability of the KG is used to exclude the non-relevant papers from the initial results set as depicted in the example query and Equation 2 (i.e., paper at position-3 didn't appear in the final results set). This exclusion of non-relevant results from the initial list improves both precision and nDCG [5], [67]. Additionally, it extends the initial results set by including relevant results from the relevant clusters identified automatically through the entity-based search from the top-*k* results, i.e., paper at position-5 included in the final list of papers list that was relevant but not appeared in the initial results list. As described in Algorithm 1, the approach obtains and returns the final more relevant results set to a user by using both *InitialRetr* and *KGRetr* through the simple model, using Equation 3.


Initial Query
Natural Language Processing Technique
Initial Retrieval
Interesting entities from top K results for Querying KG
Final Retrieval

```
"PaperID":"P14-5010",  "Relevance score : 4
"Title":["The Stanford CoreNLP Natural Language Processing Toolkit"],

"PaperID":"W05-1306",  "Relevance score : 1
"Title":["Corpus Design For Biomedical Natural Language Processing"],

"PaperID":"W02-1302",  "Relevance score : 1
"Title":["Towards A Road Map On Human Language Technology: Natural Language

"PaperID":"W06-2603",  "Relevance score : 3
"Title":["Decomposition Kernels For Natural Language Processing"],

"PaperID":"C88-2163",  "Relevance score : 2
"Title":["Default Reasoning In Natural Language Processing"],

"interesting entities ":[
   "Title:language      ",2.770207,
   "Title: Technology   ",2.6610727,
   "Title: CoreNLP      ",1.9300889,
   "Title:techniques    ",1.3301889
   "Abstract: Natural   ",3.2436814,
   "Abstract: processing",3.166624,
   "Abstract: Toolkit   ",2.7853866,
   "Abstract: Kernels    ,2.773562]}

"PaperID":"P04-3031",  "Relevance score : 4
"Title":["NLTK: The Natural Language Toolkit"],
"PaperID":"P14-3001",  "Relevance score   3
"Title":["Bayesian Kernel Methods for Natural Language Processing"],
"PaperID":"P14-5010",  "Relevance score : 4
"Title":["The Stanford CoreNLP Natural Language Processing Toolkit"],
"PaperID":"J79-1036d", "Relevance score : 4
"Title":["A Natural Language Processing Package"],
"PaperID":"W06-2603",  "Relevance score : 3
"Title":["Decomposition Kernels For Natural Language Processing"],
```

FIGURE 8. An example query.

IV. EXPERIMENTAL SETUP, EVALUATION, AND DISCUSSION

The approach is built on Apache Solr version 7.2 [68], a Lucene-based search platform, and Neo4j [69], evaluated on the ACL dataset as described in Section IV-A. For Equation 3, no complex tuning strategy is employed, rather we keep $\alpha = 0.55$, in all cases, as we have observed better results in keeping IR retrieval a little dominant. Additionally, for merging the results of both the IR and KG retrieval, we consider the top 70 results, as the dataset is comparatively smaller in size. The performance of the ISCA is assessed by comparing it with BM25, *KG_ret* while bootstrapping a full-fledged index described in Table 1.

BM25 is applied in the same way to the paper's title, abstract, body text, and metadata for comparison. The title and abstract are treated as separate fields to use different boost factors during entity retrieval and recognition. For instance,

TABLE 1. Index fields status.

	Tokenize	Stored	OTFP	Searchable
Title	✓	✓	✓	✓
Authers	✗	✓	✗	✓
Venue	✗	✓	✗	✓
Abstract	✓	✓	✓	✓
Body Text	✓	✓	✓	✓
Citations	✗	✓	✗	✓
In-cite	✗	✓	✗	✓
Out-cite	✗	✓	✗	✓
Publishers	✗	✓	✗	✓

OTFP = Omit Term Frequency and Position

the title is considered the most important followed by abstract and so on, as shown in Equation 5. Here, title, abstract, and body are binary functions that indicate the presence of the term t in the given fields. α, β and γ define the importance of each term in the respective field.

$$Freq(t)) = \alpha.Title(t) + \beta.Abstract(t) + \gamma.Body(t) \tag{5}$$

A. DATASET

For the experiments, two versions of the ACL data set were used to see the influence of the size of the data set on the search performance. It was also focused to see the difference in manual relevance judgments and the one provided by Anna Ritchie *et al.* [70]. Among the two versions of the data set, version 01 has 10,000 papers having topic queries and relevance judgments, whereas, for Version-2, we considered its 2016 dataset constructed from the papers published in Natural Language Processing (NLP) venues (journals, conferences, and workshops from 1965 to 2015) [44]. After pre-processing, version 02 of the dataset resulted in about 23058 papers for experimentation and evaluation. Table 3 presents some of the statistics about Version 02 of the dataset. The statistics about version 01 of the dataset are shown in Table 2.

TABLE 2. Statistics of the ACL Dataset-V1 [70].

Topics	Vocabulary Size	Indexed Docs	Average Relevant Docs
82	32490	9793	23.67

TABLE 3. Statistics of the ACL Dataset [44].

Network	Property	Statistics
	Papers	23058
	Authos	17695
	Venue	350
Citation Network	Nodes	17006
	Edges	103900
	Avg. Degree	11.44
Author Network	Nodes	11513
	Edges	325209
	Diameter	9
	Avg. Degree	56.49
	LCCS	11454

LCCS= Largest connected component size

A python-based abstract-extractor was implemented to efficiently extract abstracts for indexing. The Apache Tika Parser was used during the indexing phase for core content and other metadata extraction. From text to Extensible Markup Language (XML), we used our python-based script to index the metadata resourcefully in Apache Solr. The CNA was built to perform citation networks analysis by exploit incites and out-cites of papers using Equation 1. The corpus was indexed, searched, and ranked using Apache Solr and Neo4j, as described in Figure 6 and Figure 8.

B. ASSESSMENT MEASURES

Two methods have been widely used for the assessment of scholarly search systems. The first method exploits the whole reference list of a given paper to see how many of them the system can re-identify. The second method looks at the relevance judgments of human evaluators for the system's effectiveness. They check to identify whether a given paper is relevant or not to a certain query. As the first method seems somewhat prejudiced towards the CNA and may not efficiently assess the use case we have in mind. Therefore, the second method was adopted by involving human evaluators to examine the effectiveness of the proposed approach using nDCG and precision rates [6].

To demonstrate the effectiveness of ISCA in terms of nDCG [71], the top ten results were considered in computing precision over the top 5, 10, and 20 papers. For relevance judgments of the version 02 data set, the analysis of three senior Ph.D. students of Computer Science were used. These were then assessed to minimize the chances of bias and misrepresentation. In total, by following [6], the results of sixty queries were evaluated using the relevance values: 3 for highly relevant; 2 for relevant; 1 for navigational; and 0 for not relevant). Note that the same Ph.D. students were allowed to formulate these search queries. This is due to the fact that users who did the relevance judgments can produce queries that can be exploited to optimally measure their satisfaction and in evaluating results returned by the academic search engines [6]. For the relevance judgments, twenty top-ranked papers were presented to these judges in random order, where the retrieval models were kept hidden from them, we hope that such a treatment may make the judgment fair to both methods. For version-1 of the dataset, as shown in Table 2, we experimented with 82 queries and their relevance judgments provided by Anna Ritchie *et al.* [70].

C. RESULTS AND DISCUSSION OVER COMPARATIVE ANALYSIS

This section presents a comparative analysis of the proposed approach with the BM25 and *KG_ret* using standard evaluation metrics, namely precision and nDCG. Overall, the framework is analyzed from three perspectives. First, we evaluate the effectiveness of the entire full-fledged hybrid index to see the influence of KG in retrieving relevant articles. In this case, we evaluate both versions of the dataset. Second, we compare the proposed approach with the recently

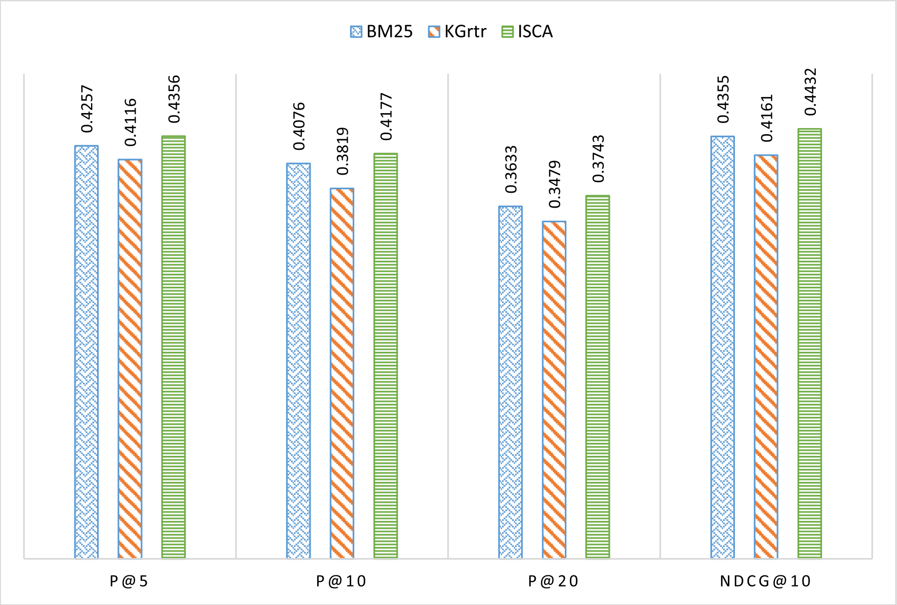


FIGURE 9. BM25, ISCA and KGrtr comparison using *Precision@P* and *nDCG@P*, while experimenting ACL Dataset V-2.

proposed technique called Real-time Feedback Query Expansion (RTFQE) [6] and observe the influence of the freshness in academic search. The RTFQE consists of two-step, in the first step the initial results set is produced and in the second step, the expanded query is formulated for the final retrieval. Finally, we perform some micro-analysis at the query level to examine the effectiveness of the proposed hybrid approach. All these perspectives are described in the following three cases.

1) CASE-I

Table 4 and Figure 9 summarize the results of version 02 of the dataset having 23058 papers by computing $nDCG@10$ and $P@5$, $P@10$, and $P@20$ using the analysis of human evaluators. The top 20 results for each query were presented in random order, and the retrieval models were kept hidden from them (as described in Section IV-B) during judgment. The quires were distributed as per their subject areas to achieve fair judgment for analysis, using the aforementioned 4-point scale formula. Each student has evaluated the result of 60 queries (each student evaluates, i.e., $60 * 20 = 1200$ papers). In total, 3600 papers were evaluated for sixty different queries of BM25, KGrtr, and proposed ISCA.

Using human evaluators, 43% of the papers were judged as highly relevant, 17% as relevant (i.e., fair), 20% navigational, and 20% non-relevant. The results in Table 4 and Figure 9 show that ISCA performs better than both BM25 and KGrtr, especially at $P@5$, $P@10$, and $nDCG@10$. The analysis of the first experiment shows that higher precision can be obtained at the top five of the ranked results. Likewise, the comparative analysis on version 01 of the dataset with 82 queries and relevance judgments is shown in Figure 10.

TABLE 4. Performance comparison of ISCA with BM25 and KGrtr.

Method	$P@5$	$P@10$	$P@20$	$nDCG@10$
BM25	0.4257	0.4076	0.3633	0.4432
KGrtr	0.4116	0.3819	0.3479	0.4161
ISCA	0.4356	0.4177	0.3743	0.4432

Figures 9 and 10 also demonstrate that as the size of the dataset increases, the performance of the proposed approach increases. We believe that better performance could be achieved with larger datasets.

2) CASE-II (COMPARISON WITH RTFQE [6] AND IMPACT OF RECENCY FACTOR IN ACADEMIC SEARCH)

The RTFQE technique [6] practices query expansion to support the academic search. The approach uses BM25 and citation analysis for retrieving the first results list in the same way as ISCA. Then the entities are extracted from the top-k results of the initial results set and run it against the KG using the clustering and VSM capability of Neo4j to refine the search results, as discussed in Section III. In contrast, RTFQE [6] extracts interesting terms and runs it against the same index as relevance feedback by combining query expansion and CNA. Table 5 demonstrates the comparative analysis (i.e., for this we used version 01 of the dataset having relevance judgments of about 82 queries) [6].

As we can see in Table 5, ISCA performs well at both the top 5 and 10 results and close in performance to RTFQE at the top 20 results. We perform this analysis only on 23058 articles of the ACL dataset and believe that the combination of both the structure (KG) and the inverted index, along with citation analysis can better meet the needs of scholarly users. In the

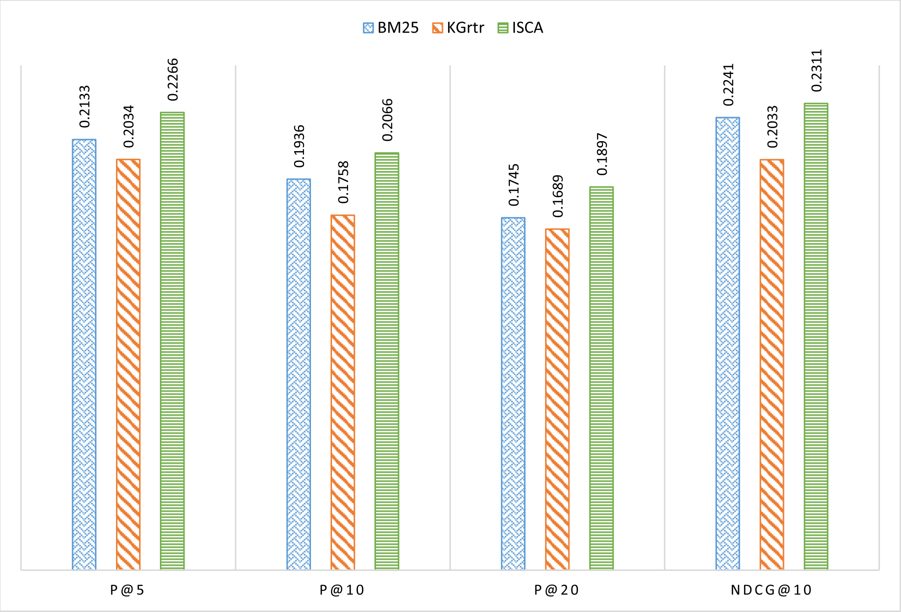


FIGURE 10. BM25, ISCA and KGrtr comparison using *Precision@P* and *nDCG@P*, while experimenting ACL Dataset V-1.

TABLE 5. Performance comparison of RTFQE [6] and ISCA.

Method	P@5	P@10	P@20	nDCG@10
RTFQE	0.2164	0.1933	0.1812	0.2271
ISCA	0.2266	0.2067	0.1887	0.2311

academic search, scholarly paper's recency/freshness can be considered as one of the essential factors [2]. Besides, as a scholarly repository being rapidly growing, it is essential to offer the intended fresh and relevant documents to academic searchers. Therefore, this hybrid ranking technique gives more weight to fresh and relevant articles to see the impact of freshness during analysis. For recency, first, we identify, compute, and categorize the freshness factor of each paper in the given ACL collection using publication year, as shown in Table 6.

TABLE 6. Yearly distribution of published articles.

Interval	Number of Papers
1965 to 1975	257
1975 to 1985	862
1985 to 1995	3083
1995 to 2005	5671
2005 to 2014	13176

Keeping in view the above interval, we integrated the freshness factor by defining a recency factor as $R_f(P_i) = 1/(y^{1/a})$. Here, R_f is the recency factor of P_i, and a is a parameter with values such as 1, 2, 3, and so on. The experimental results after including the normalized recency factor are shown in Figure 11. To study the influence of the recency on the scoring scheme, we tuned the recency component and compared the results of changes in the scoring system. For example, if the recency is raised to the power three, its value is weighted three times more than the other scoring components. On the other hand, if the power is zero, the recency factor is not used at all in the scoring. As described in Table 7, when we give more weight to the recency factor, papers

TABLE 7. Search results while boosting recency factor.

Rank	Tuning Recency			
	0	1.0	2.0	3.0
$1_{(2000)}$	A00-1012	A00-1012	A00-1012	A00-1012
$2_{(1983)}$	A83-1019	A83-10199	A83-1019	A83-1019
$3_{(1992)}$	A92-1007	A92-1007	A92-1007	A92-1007
$4_{(2008)}$	I08-2097	P14-1117	P14-1117	P14-1117
$5_{(2005)}$	I05-1001	I05-1001	I05-1001	P14-5014
$6_{(2013)}$	A013-4034	A013-4034	A013-4034	A013-4034
$7_{(2014)}$	P14-1117	I08-2097	I08-2097	I08-2097
$8_{(2002)}$	W02-1816	W02-1816	P14-5014	I05-1001
$9_{(1979)}$	P79-1006	P79-1006	P79-1006	W02-1816
$10_{(2014)}$	P14-5014	P14-5014	W02-1816	P79-1006

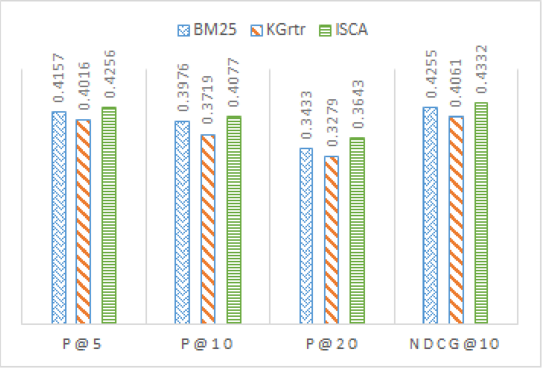


FIGURE 11. Performance comparison including recency factor (R_f).

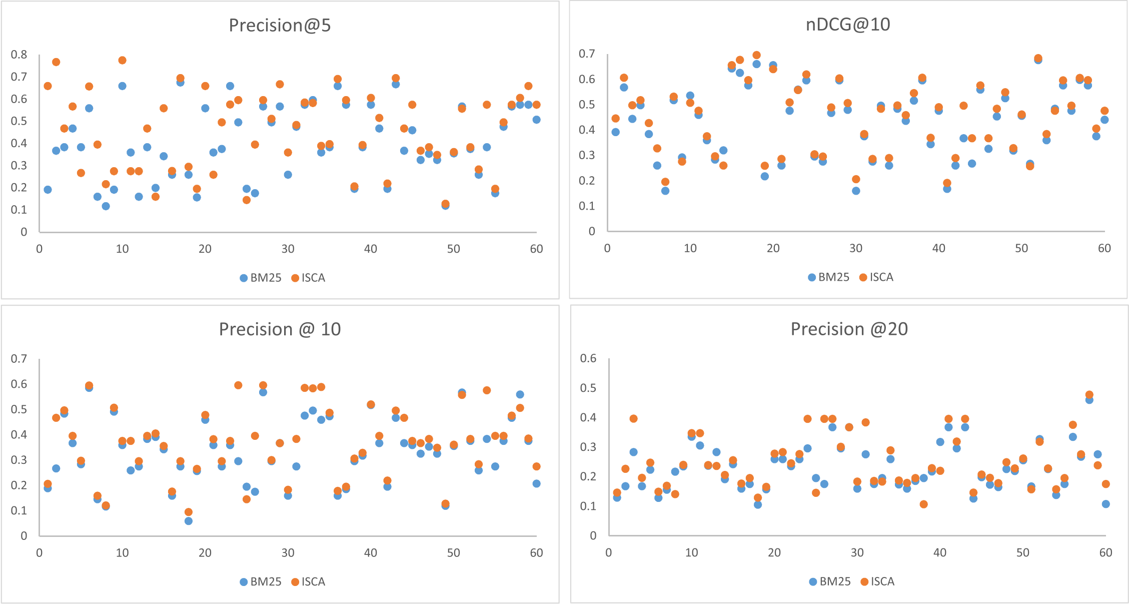


FIGURE 12. Performance analysis of sixty queries by using Precision@P and nDCG@P.

that have been published more recently, e.g., P14-1117, P14-5014 at position seven and eight are periodically getting to the top in ranking. On the other hand, like A83-1019 at the second position, which is older comparatively is going down while tuning recency factor. This result supports our argument that the recency factor is useful to a user who wants to get up-to-date knowledge. We proposed a tunable scoring function, so users can customize the weights of scoring components according to their search purposes. Thanks to recency factor (R_f), the statistics and user relevance judgments delineated in Figure 11 show that academic searchers prefer fresh and relevant articles.

3) CASE-III (MICRO ANALYSIS)

We also analyzed the grading of 60 queries and compared ISCA with BM25 that uses only an inverted index. At Precision@5 in Figure 12, the orange dots of ISCA dominate most of the blue dots (which represent BM25), indicating that the proposed approach performs well at the top 5 results. Also, the performance of ISCA and BM25 is closed at the top 20. Figure 12 presents a more detailed view of this situation. We can see in Figure 9, Figure 10, and Figure 12 that ISCA gives significantly better results at the top 5, 10 and 20 results, as given in the fourth row of Table 4 and third row of Table 5. The average performance of ISCA by Precision@5 is 0.4356, which is greater than that of BM25 (0.4257), and KGrtr (0.4116) at top results. For a majority of cases (66%) at Precision@5, ISCA outperformed BM25. Likewise BM25 (precision@5) has greater standard deviation (0.1651) than that of ISCA (0.1636). The statistical analysis illustrates the improvement in terms of the ACL dataset. This analysis illustrates that the hybrid approach of combining classical IR and CNA in the form of ISCA can rank papers more effectively in scholarly search by considering both the textual and structural information for ranking. From the analysis on both versions of the dataset, it can be concluded that the performance gap of both the ISCA and baseline models increases as the size of the dataset increases.

The main objective of this research is to practice and evaluate the combination of KG and CNA with the IR model in the academic search. Some key observations of this research work can be concluded as (a) both KG and IR model with CNA can improve academic search. (b) The efficient extraction and indexing of title, abstract, full-content, metadata, and their relationship along with citation networks can complement ranking more effectively by expanding the user keywords query through the top-*k* entities from the initial results list and exploiting it over KG for final retrieval.

V. CONCLUSION

Scholarly search engines aim to ease the manual effort of academic searchers in discovering the relevant publications against the search queries while in the quest of finding answers to certain research questions. However, the considerable expansion and complexity of the scholarly document collections make the academic search an interesting yet challenging area to explore. Numerous solutions have been developed to alleviate this issue in order to enable users to bring more relevant and intended results against their information need represented in the form of a search query.

In the last decade, academic search has attracted several renowned research teams from prestigious institutions and research centers/labs around the world. Their solutions have been successful up to a greater extent, yet there are always chances for improvement. In this regard, this research work is an attempt to improve the search experience of researchers and scholars by developing, presenting, and evaluating a hybrid approach for academic search. The solution proposed, termed as ISCA, uses KG (structured search), classical IR, and CNA to support academic search. Besides, it uses LDA in identifying the topics of papers to build a structural search system for discovering the propagation of topics in the citation network. The mechanism includes building KG, which is suitable for connected data with citation networks. The evaluation illustrates that the framework can filter the most relevant scholarly papers in a more nuanced way, comparatively. The experimental results also demonstrate that the use of structured search on top of IR models can improve the strategic ranking of academic search engines and lead to significantly better results.

The presented solution can be extended in several ways. In the near future, we would like to consider other factors such as user query log and academic profile to enrich KG for optimal entities extraction to rank the top most relevant papers. One can also practice the efficiency of this hybrid approach in a distributed environment. Other data fusion methods may also be applied to combine the results of both retrieval models.

LIST OF ABBREVIATIONS

AAN	ACL Anthology Network.
ACL	Association for Computational Linguistics.
CN	Citation Network.
CNA	Citation Networks Analysis.
IR	Information Retrieval.
ISCA	Inverted Indices and Structure Search with Citation Analysis.
KG	Knowledge Graph.
KGrtr	Knowledge Graph Retrieval.
LDA	Latent Dirichlet Allocation.
LSA	Latent Semantic Analysis.
nDCG	Normalized Discounted Cumulative Gain.
NLP	Natural Language Processing.
PRBL	Processing and Retrieval Business Logic.
RTFQE	Real-time Feedback Query Expansion.
S2	Semantic Scholar.
SDKC	Scholarly Data and Knowledge Curation.
TF	Term Frequency.
TV	Topic Vector.
UI	User Interface.
VSM	Vector Space Model.
XML	Extensible Markup Language.

REFERENCES

[1] N. Fiorini, D. J. Lipman, and Z. Lu, "Cutting edge: Towards PubMed 2.0," *Elife*, vol. 6, Oct. 2017, Art. no. e28801.

[2] S. Lee, D. Kim, K. Lee, J. Choi, S. Kim, M. Jeon, S. Lim, D. Choi, S. Kim, A.-C. Tan, and J. Kang, "BEST: Next-generation biomedical entity search tool for knowledge discovery from biomedical literature," *PLoS ONE*, vol. 11, no. 10, Oct. 2016, Art. no. e0164680.

[3] S. Khalid, S. Khusro, I. Ullah, and G. Dawson-Amoah, "On the current state of scholarly retrieval systems," *Eng., Technol. Appl. Sci. Res.*, vol. 9, no. 1, pp. 3862–3869, 2019.

[4] A. Martín-Martín, E. Orduna-Malea, M. Thelwall, and E. D. López-Cózar, "Google scholar, web of science, and scopus: A systematic comparison of citations in 252 subject categories," *J. Informetrics*, vol. 12, no. 4, pp. 1160–1177, Nov. 2018.

[5] O. A. Abass, O. Folorunso, and B. O. Samuel, "Automatic query expansion for information retrieval: A survey and problem definition," *Amer. J. Comput. Sci. Inf. Eng.*, vol. 4, no. 3, pp. 24–30, 2017.

[6] S. Khalid, S. Wu, A. Alam, and I. Ullah, "Real-time feedback query expansion technique for supporting scholarly search using citation network analysis," *J. Inf. Sci.*, vol. 47, no. 1, 2019, Art. no. 0165551519863346.

[7] R.-L. Liu, "Retrieval of scholarly articles with similar core contents," *Int. J. Knowl. Content Develop. Technol.*, vol. 7, no. 3, pp. 5–27, 2017.

[8] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1271–1279.

[9] D. Mirylenka, "Towards structured representation of academic search results," Ph.D. dissertation, School Inf. Commun. Technol., Univ. Trento, Trento, Italy, 2015.

[10] M. Kluck and M. Stempfhuber, "Domain-specific track CLEF 2005: Overview of results and approaches, remarks on the assessment analysis," in *Proc. Workshop Cross-Lang. Eval. Forum Eur. Lang.* Berlin, Germany: Springer, 2005, pp. 212–221.

[11] J. L. Ortega, *Academic Search Engines: A Quantitative Outlook*. Amsterdam, The Netherlands: Elsevier, 2014.

[12] S. Khalid and S. Wu, "Supporting scholarly search by query expansion and citation analysis," *Eng., Technol. Appl. Sci. Res.*, vol. 10, no. 4, pp. 6102–6108, Aug. 2020.

[13] A. Rattinger, J. Le Goff, and C. Guetl, "Local word embeddings for query expansion based on co-authorship and citations," in *Proc. 7th Int. Workshop Bibliometric-Enhanced Inf. Retr. (BIR), 40th Eur. Conf. Inf. Retr. (ECIR)*, Grenoble, France, vol. 2080, 2018, pp. 46–53.

[14] B. Golshan, T. Lappas, and E. Terzi, "Sofia search: A tool for automating related-work search," in *Proc. 2012 ACM SIGMOD Int. Conf. Manage. Data*, 2012, pp. 621–624.

[15] M. Hagen, A. Beyer, T. Gollub, K. Komlossy, and B. Stein, "Supporting scholarly search with keyqueries," in *Proc. Eur. Conf. Inf. Retr.* Padua, Italy: Springer, 2016, pp. 507–520.

[16] X. Li, B. J. A. Schijvenaars, and M. de Rijke, "Investigating queries and search failures in academic search," *Inf. Process. Manage.*, vol. 53, no. 3, pp. 666–683, May 2017.

[17] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proc. 2nd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2009, pp. 5–14.

[18] S. Shogen, T. Shimizu, and M. Yoshikawa, "Enrichment of academic search engine results pages by citation-based graphs," in *Proc. AIRS*. Brisbane, QLD, Australia: Springer, Dec. 2015, pp. 56–67.

[19] T. Khazaei and O. Hoeber, "Supporting academic search tasks through citation visualization and exploration," *Int. J. Digit. Libraries*, vol. 18, no. 1, pp. 59–72, Mar. 2017.

[20] S. Salehi, J. T. Du, and H. Ashman, "Examining personalization in academic web search," in *Proc. 26th ACM Conf. Hypertext Social Media (HT)*, 2015, pp. 103–111.

[21] D. Wu, S. Liang, and W. Yu, "Collaborative information searching as learning in academic group work," *Aslib J. Inf. Manage.*, vol. 70, no. 1, pp. 2–27, Jan. 2018.

[22] S. Khalid, S. Wu, and F. Zhang, "A multi-objective approach to determining the usefulness of papers in academic search," *Data Technol. Appl.*, vol. 55, no. 3, pp. 1–15, May 2021. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/DTA-05-2020-0104/full/pdf, doi: 10.1108/DTA-05-2020-0104.

[23] T. Nguyen and P. Do, "Managing and visualizing citation network using graph database and LDA model," in *Proc. 8th Int. Symp. Inf. Commun. Technol.*, Berlin, Germany, Dec. 2017, pp. 100–105.

[24] A. Tonon, G. Demartini, and P. Cudré-Mauroux, "Combining inverted indices and structured search for ad-hoc object retrieval," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2012, pp. 125–134.

[25] G. Mai, K. Janowicz, and B. Yan, "Combining text embedding and knowledge graph embedding techniques for academic search engines," in *Proc. Semdeep/NLIWoD@ ISWC*, 2018, pp. 77–88.

[26] H. Amiri, A. AleAhmad, F. Oroumchian, C. Lucas, and M. Rahgozar, "Using OWA fuzzy operator to merge retrieval system results," Univ. Tehran, Tehran, Iran, Tech. Rep., 2007. [Online]. Available: https://ro.uow.edu.au/cgi/viewcontent.cgi?referer=https://scholar.google.com.pk/&httpsredir=1&article=1006&context=dubaipapers

[27] A. Singhal, "Introducing the knowledge graph: Things, not strings," in *Official Google Blog*, vol. 16. Wilmette, IL, USA: Benton Institute for Broadband & Society, 2012. [Online]. Available: https://www.benton.org/headlines/introducing-knowledge-graph-things-not-strings

[28] J. Wu, J. Killian, H. Yang, K. Williams, S. R. Choudhury, S. Tuarob, C. Caragea, and C. L. Giles, "Pdfmef: A multi-entity knowledge extraction framework for scholarly documents and semantic search," in *Proc. 8th Int. Conf. Knowl. Capture*, 2015, pp. 1–8.

[29] R. A. Al-Zaidy and C. L. Giles, "Automatic knowledge base construction from scholarly documents," in *Proc. ACM Symp. Document Eng.*, New York, NY, USA, Aug. 2017, pp. 149–152, doi: 10.1145/3103010.3121043.

[30] M. Singh, B. Barua, P. Palod, M. Garg, S. Satapathy, S. Bushi, K. Ayush, K. Sai Rohith, T. Gamidi, P. Goyal, and A. Mukherjee, "OCR++: A robust framework for information extraction from scholarly articles," 2016, *arXiv:1609.06423*. [Online]. Available: http://arxiv.org/abs/1609.06423

[31] R. A. Al-Zaidy and C. L. Giles, "Automatic extraction of data from bar charts," in *Proc. 8th Int. Conf. Knowl. Capture*, Oct. 2015, pp. 1–4.

[32] X. Lu, S. Kataria, W. J. Brouwer, J. Z. Wang, P. Mitra, and C. L. Giles, "Automated analysis of images in documents for intelligent document search," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 12, no. 2, pp. 65–81, Jul. 2009.

[33] S. Tuarob, S. Bhatia, P. Mitra, and C. L. Giles, "AlgorithmSeer: A system for extracting and searching for algorithms in scholarly big data," *IEEE Trans. Big Data*, vol. 2, no. 1, pp. 3–17, Mar. 2016.

[34] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a web of open data," in *The Semantic Web*. Brisbane, QLD, Australia: Springer, 2007, pp. 722–735.

[35] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[36] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2012, pp. 481–492.

[37] A. Martín-Martín, M. Thelwall, E. Orduna-Malea, and E. D. López-Cózar, "Google scholar, Microsoft academic, scopus, dimensions, web of science, and opencitations' COCI: A multidisciplinary comparison of coverage via citations," *Scientometrics*, vol. 126, no. 1, pp. 871–906, 2021.

[38] J. Wu, K. M. Williams, H.-H. Chen, M. Khabsa, C. Caragea, S. Tuarob, A. G. Ororbia, D. Jordan, P. Mitra, and C. L. Giles, "CiteSeerX: AI in a digital library search engine," *AI Mag.*, vol. 36, no. 3, pp. 35–48, Sep. 2015.

[39] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 990–998.

[40] Z. Lu, "PubMed and beyond: A survey of web tools for searching biomedical literature," *Database*, vol. 2011, no. 2011 baq036, pp. 1–13, Jan. 2011.

[41] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and K. Wang, "An overview of Microsoft academic service (MAS) and applications," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 243–246.

[42] J. Shen, Z. Song, S. Li, Z. Tan, Y. Mao, L. Fu, L. Song, and X. Wang, "Modeling topic-level academic influence in scientific literatures," in *Proc. AAAI Workshop: Scholarly Big Data*, 2016, pp. 1–7.

[43] X. Ren, J. Shen, M. Qu, X. Wang, Z. Wu, Q. Zhu, M. Jiang, F. Tao, S. Sinha, D. Liem, P. Ping, R. Weinshilboum, and J. Han, "Life-iNet: A structured network-based knowledge exploration and analytics system for life sciences," in *Proc. ACL Syst. Demonstrations*, 2017, pp. 55–60.

[44] D. R. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara, "The ACL anthology network corpus," *Lang. Resour. Eval.*, vol. 47, no. 4, pp. 919–944, 2013.

[45] R. A. Al-Zaidy and C. L. Giles, "Extracting semantic relations for scholarly knowledge base construction," in *Proc. IEEE 12th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2018, pp. 56–63.

[46] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab, "Learning taxonomic relations from heterogeneous sources of evidence," *Ontol. Learn. Text: Methods, Eval. Appl.*, vol. 123, pp. 59–73, Jul. 2005.

[47] J. Guo, G. Xu, X. Cheng, and H. Li, "Named entity recognition in query," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2009, pp. 267–274.

[48] J. S. Whissell and C. L. A. Clarke, "Effective measures for inter-document similarity," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 1361–1370.

[49] K. W. Boyack, D. Newman, R. J. Duhon, R. Klavans, M. Patek, J. R. Biberstine, B. Schijvenaars, A. Skupin, N. Ma, and K. Börner, "Clustering more than two million biomedical publications: Comparing the accuracies of nine text-based similarity approaches," *PLoS ONE*, vol. 6, no. 3, Mar. 2011, Art. no. e18029.

[50] P. Glenisson, W. Glänzel, F. Janssens, and B. De Moor, "Combining full text and bibliometric information in mapping scientific disciplines," *Inf. Process. Manage.*, vol. 41, no. 6, pp. 1548–1572, Dec. 2005.

[51] T. K. Landauer, D. Laham, and M. Derr, "From paragraph to graph: Latent semantic analysis for information visualization," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 1, pp. 5214–5219, Apr. 2004.

[52] M. Cambridge, "Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track," in *Proceedings*. Gaithersburg, MD, USA: NIST Special Publication 500-242, National Institute of Standards and Technology (NIST), 1999.

[53] L. Egghe and R. Rousseau, *Introduction to Informetrics: Quantitative Methods in Library, Documentation and Information Science*. Amsterdam, The Netherlands: Elsevier, 1990.

[54] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[55] X. Yan, P. S. Yu, and J. Han, "Substructure similarity search in graph databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2005, pp. 766–777.

[56] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou, "EASE: An effective 3-in-1 keyword search method for unstructured, semi-structured and structured data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 903–914.

[57] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional expansion for keyword search on graph databases," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 505–516.

[58] K. Golenberg, B. Kimelfeld, and Y. Sagiv, "Keyword proximity search in complex data graphs," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2008, pp. 927–940.

[59] H. Raviv, O. Kurland, and D. Carmel, "Document retrieval using entity-based language models," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 65–74.

[60] F. Zhang and S. Wu, "Ranking scientific papers and venues in heterogeneous academic networks by mutual reinforcement," in *Proc. 18th ACM/IEEE Joint Conf. Digit. Libraries*, May 2018, pp. 127–130.

[61] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., 1999.

[62] A. Bellogín and A. Said, "Information retrieval and recommender systems," in *Data Science in Practice*. Cham, Switzerland: Springer, 2019, pp. 79–96. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-97556-6_5

[63] S. Wu and S. McClean, "Performance prediction of data fusion for information retrieval," *Inf. Process. Manage.*, vol. 42, no. 4, pp. 899–915, 2006.

[64] S. Wu and F. Crestani, "Data fusion with estimated weights," in *Proc. 11th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2002, pp. 648–651.

[65] I. Rasheed, H. Banka, and H. M. Khan, "Pseudo-relevance feedback based query expansion using boosting algorithm," *Artif. Intell. Rev.*, pp. 1–24, Feb. 2021, doi: 10.1007/s10462-021-09972-4.

[66] J. L. da Silva Devezas, "Graph-based entity-oriented search," Univ. Porto, Porto, Portugal, Tech. Rep., 2021. [Online]. Available: https://repositorio-aberto.up.pt/bitstream/10216/133205/2/450176.pdf

[67] Y. Lv and C. Zhai, "Positional relevance model for pseudo-relevance feedback," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2010, pp. 579–586.

[68] T. Grainger and T. Potter, *Solr in Action*. Shelter Island, NY, USA: Manning, 2014.

[69] A. Vukotic, N. Watt, T. Abedrabbo, D. Fox, and J. Partner, *Neo4j Action*, vol. 22. Shelter Island, NY, USA: Manning, 2015.

[70] B. H. A. Ritchie, "Bob, the ACL anthology test collection," Ph.D. dissertation, Comput. Lab., Univ. Cambridge, Cambridge, U.K., 2017.

[71] 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. [Online]. Available: https://doi.org/10.1145/582415.582418

SHAH KHALID received the M.S. degree in computer science from the Department of Computer Science, University of Peshawar, Pakistan. He is currently pursuing the Ph.D. degree with the School of Computer Science and Communication Engineering, Jiangsu University, China. His research interests include information retrieval, web search engines, scholarly retrieval systems, recommender systems, knowledge graphs, social web, web engineering, and digital libraries. He has been doing teaching and research with the National University of Science and Technology (NUST), Islamabad, Pakistan. He has been involved in many research projects in Pakistan and abroad.

SHENGLI WU (Member, IEEE) received the Ph.D. degree in computer science from Southeast University, China. He is currently a Lecturer with the School of Computing, Ulster University, U.K. His current research interests include database and information retrieval, scientometrics and citation analysis, and machine learning.

ABDUL WAHID received the Ph.D. degree in engineering from Kyungpook National University, Daegu, Republic of Korea. He is currently an Assistant Professor with the School of Computing, National University of Sciences and Technology, Islamabad, Pakistan. His current research interests include SDN, VANET, Underwater Sensor Network, routing protocols, wireless sensor networks, network protocols, and sensors.

AFTAB ALAM received the B.S. degree from the University of Malakand, Khyber Pakhtunkhwa, Pakistan, the M.S. degree from the University of Peshawar, Khyber Pakhtunkhwa, and the Ph.D. degree from Kyung Hee University, South Korea, all in computer science. He is currently working as a Researcher with the Department of Computer Science and Engineering, Kyung Hee University (Global Campus). His research interests include big data analytics in the cloud, distributed computing, information retrieval, social computing, machine/deep learning, information service engineering, knowledge engineering, complex event analysis in the multi-stream environment, and graph mining.

IRFAN ULLAH received the B.S. degree in computer science from the Department of Computer Science, University of Malakand, Pakistan, and the M.S. and Ph.D. degrees in computer science from the Department of Computer Science, University of Peshawar, Pakistan. He is currently working as an Assistant Professor with the Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal, Pakistan. He has more than ten years of teaching and research experience. He is the author of more than 30 research papers published in journals and conferences of international repute. His research interests include information retrieval, interactive information retrieval, information service engineering, web semantics, linked and open data, ontology engineering, social web, and social book search.