Chapter 3

DISCOVER: A Sequential Approach-based Academic Venue Recommender System

"All life is an experiment. The more experiments you make the better."

-Ralph Waldo Emerson (1803-1883)

3.1 Introduction

Researchers generally favour to publish in those academic venues (journals, conferences, or workshops) where they find acknowledgement of top notch papers applicable to their domain of research [150]. However, with swift evolution and expansion of domains of multidisciplinary areas, there has been an active change within the range of journals, henceforth making the choice of an appropriate venue an even more burdensome task [36]. Although a researcher may know a few leading high-profile venues for her specific field of interest, a venue recommender system becomes particularly helpful when one explores a new field or when more options are needed.

We propose DISCOVER: A Diversified yet Integrated Social network analysis and COntextual similarity-based scholarly VEnue Recommender system. Our work provides an integrated framework incorporating social network analysis, including centrality measure calculation, citation and co-citation analysis, topic modeling based contextual similarity and main path analysis of a bibliographic citation network.

3.2 Problem Description

Let $G = (V, E)$ be a citation graph with n papers, such that $V = \{p_1, p_2, ..., p_n\}$, and each directed edge $e = (p_i, p_j) \in E$ represents a citation from paper p_i to p_j. We use the following two phrases to describe the citation network.

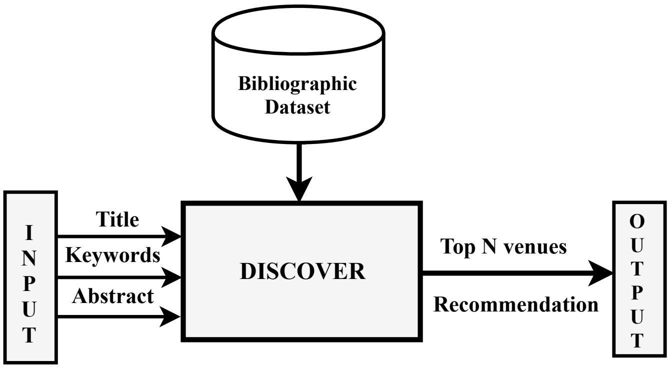


Figure 3.1: Overview of DISCOVER

(i) References of p_i represent the set of papers which are referred by the paper p_i.

(ii) Citation to p_j denotes the set of papers which have used the paper p_j as a reference.

For the rest of the paper, we use the above two phrases to define the graph around vertex p_i. Let each paper p_i be published in a particular venue v_i. So now we have, $S = \{v_1, v_2, ..., v_n\}$ be a predefined set of publication venues (not all v_i's are necessarily unique). Given an input paper (seed paper) p_0, the venue recommendation task is to recommend an ordered list of suitable publication venues (v_{0_1}, v_{0_2} ,..., v_{0_k}) related to the seed paper p_0, such that v_{0_1} is the most relevant and v_{0_k} is k-th most relevant venue in the decreasing order of relevance or suitability.

Hence it is primarily a ranking problem. We need first to figure out the set of papers which are closely related to the seed paper and then rank them. Venue recommendations are provided if the title, keywords, and abstract of a seed paper are given to the system as input (Fig. 3.1).

3.3 The Functional Architecture of DISCOVER

We introduce the overall architecture of the proposed system DISCOVER with its operational methods. DISCOVER is designed for shortlisting academic venues to make a personalized recommendation for researchers. It has a layered approach where each layer performs a specific task used by the next layer.

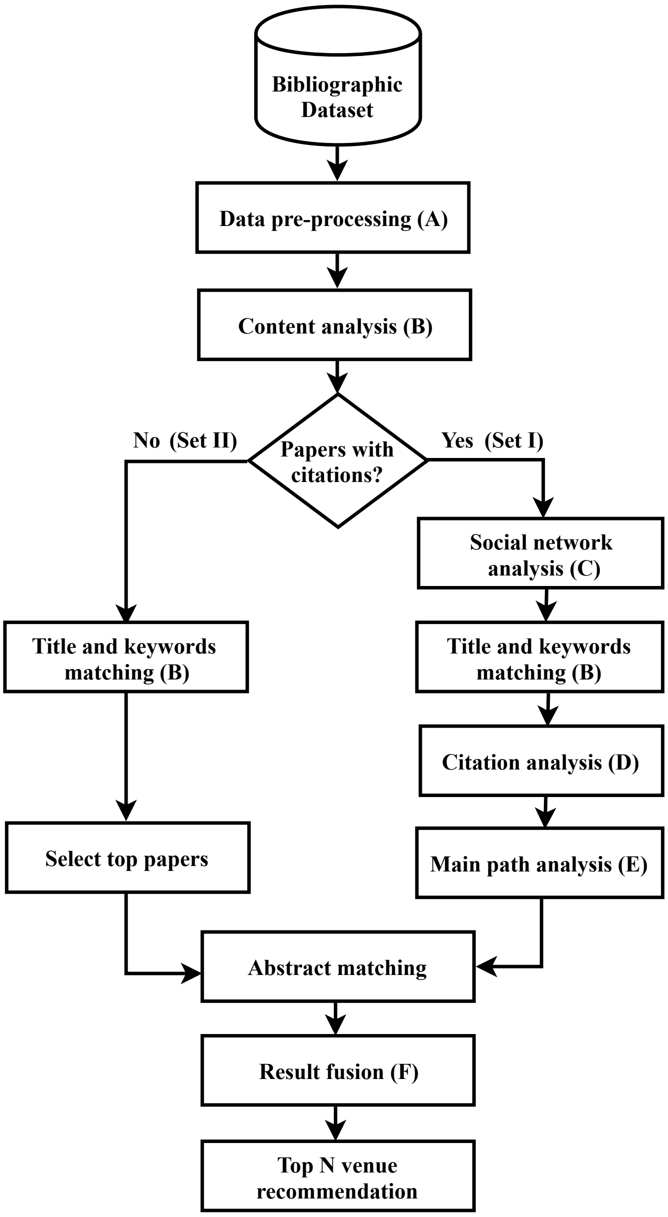


Figure 3.2: Organizational architecture of DISCOVER

3.3.1 Framework of DISCOVER

DISCOVER is based on social network analysis where the association of nodes in networks and the significance of individual nodes are considered. The overall process comprises the following six steps (Fig. 3.2).

A. ***Data Preprocessing***: This step aims to structure, arrange, and organize the dataset suitable for faster extraction of relevant papers.

B. ***Content Analysis (Field of Study, Keyword, Title, Abstract Matching)***: This module is introduced to filter relevant papers based on fields of study, keywords, title, and/or abstract matching. This step may be utilized multiple times at different point of time when one or more types of matching are done.

C. ***Social Network Analysis***: Various centrality measures like degree, closeness, betweenness, eigenvector and HITS score, etc. are calculated of the papers shortlisted in previous Steps A and Step B that will be used down the line.

D. ***Citation Analysis***: This module is used to accomplish two objectives. First, identification of the most similar papers as papers of interest (I) with the help of title, keyword and abstract similarity. Then, by applying bibliographic coupling (BC), co-citation scores(CC) and a new distance measure, the most related papers to the paper of interest (I) are selected.

E. ***Main Path Analysis***: To determine the most influential papers in the citation network, traversal counts like search path count (SPC) is used. Key-route search is employed to select significant links during both local and global search to identify the global key-routes.

F. ***Result Fusion***: The final ranking of scholarly venues is done based on abstract similarity using LDA and NMF. A score based fusion technique (combMNZ) is applied to leverage the advantages of both methods.

The details of the above steps are described below.

3.3.2 A. Data Preprocessing

The original MAG dataset is organized in a hierarchical fashion divided into 4 levels (Level 0 being the root and Level 3 the leaves) where levels correspond to the field of study. Levels are related in super class - sub class relation where lower levels are subsumed in upper levels. But, a paper belonging to a Level-3 node can be a part of multiple Level-2 nodes (in case of inter-disciplinary fields), and, following the same logic, of multiple Level-1 nodes. Hence, locating the field of study to get all the relevant papers using only the keywords is not very straight-forward, but often can be very tedious and time-intensive.

The dataset, therefore, is reorganized using a hybrid binary tree. Fig. 3.3 illustrates the modifications done. We use the relation between the fields of study (FOS) in the original graph. FOSs related to each other are provided with a pair-wise confidence score based on their similarity. A score of 1 implies that the two fields are very similar (part of or dependent on each other) and a lower score implies lesser similarity. If the two fields are not similar at all, their confidence score will be 0. We use confidence scores among FOSs to divide them into two groups of children nodes at each level, one greater than the average confidence score (left children) and the other equal to and less than (right children) the average confidence score of all the children nodes with the parent node (FOS).

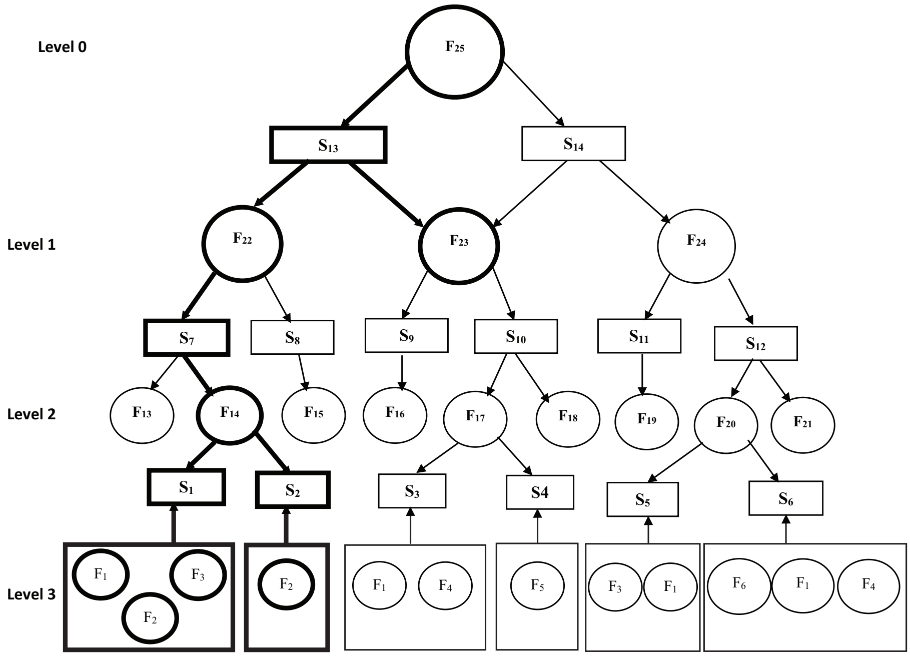


Figure 3.3: Hybrid binary tree to hierarchy of field of studies

For example, field of study F_{22}, which is at Level-1 has three children's like F_{13}, F_{14} and F_{15}. These children are divided into two groups:

(i) Field of studies F_{13} and F_{14} have a high confidence score with the field of study F_{22} and are placed on the left-hand side of the field of study F_{22}.

(ii) The field of study F_{15} shows a less confidence score with the field of study F_{22} and are placed on the right-hand side of the field of study F_{22}.

Keyword-set Construction and Organization

Keywords are identified as available under the keywords tag of research papers. Stopwords, if any, are removed and keywords are stemmed using Snowball stemmer [151]. The keywords of all the papers are fetched in a bottom-up fashion, and their union is stored at between two levels of FOS (rectangular boxes in Fig. 3.3). For example, the keyword-set between Level-3 and Level-2 (for example, S_1) is constructed by concatenating the keywords of the field of study at Level-3 (F_1, F_2, and F_3). Similarly, keyword-set between Level-2 and Level-1 (S_7) is constructed by concatenating the keywords from papers in Level-2 FOSs, i.e. (F_{13}, and F_{14}).

3.3.3 B. Content Analysis (Keyword-based Search Strategy)

We traverse the above tree in a top-down fashion to search for papers. We extract only those papers having high similarity with the keywords of a given seed paper. A queue Z is created and maintained to keep track of the visited nodes. At the start, Z contains only the root nodes corresponding to the field of study (e.g., F_{25}). A node is popped from Z, and the given set of keywords is matched with its (popped node) left, right and parent sets of keywords of the popped node separately. Upon a match, the number of matches is checked and proceeded in the following way.

(i) *Case 1*: If the number of matches is greater on any one side (either left or right), the other side is ignored. All the nodes of the greater side are added to queue Z.

(ii) *Case 2*: If the number of matches on both sides is equal, nodes from both the sides are added to the queue Z.

(iii) *Case 3*: If the number of matches of the parent is equal to that of the greater side or all three are equal, even the parent node is added to the queue Z.

This process is repeated until we reach leaf level (or Level-3). There is a duplication of data in the proposed hybrid binary tree, but the computation time is enormously reduced as at every step, just like binary search trees, the unmatched half side of the tree is not considered.

Illustrative Example of Keyword-based Search

Suppose we are matching the keywords for a given paper p_m. Initially, we have only one field of study, F_{25} in the queue Z. This node is popped, and keywords of p_m are matched with the keyword sets at level 0, namely S_{13} and S_{14} and the parent field keywords (F_{25}). Let the left subtree have a clear maximum number of matches. We then proceed in that direction and successively push F_{22} and F_{23} into the queue Z. For each of these nodes, the search is done similar to the that done for F_{25}. Finally, when we reach Level 3, now, we have all the relevant fields of study ids in our queue, including the primitive lower level keywords and some higher-level keywords matching with the given keywords.

Suppose at Level 3, only fields of study like F_1, F_2, F_3 and F_{14} are left in the queue. In Fig. 3.3, all papers belonging to fields F_1, F_2, F_3 and F_{14} are fetched and used for further analysis.

For the papers so selected, the following procedures are adopted.

3.3.4 C. Social Network Analysis

Depending on the availability of citations, the shortlisted papers will be divided as follows (Fig. 3.2).

(a) all papers whose citations exist in the dataset (Set-I)

(b) the set of papers whose citations are not available in the dataset (Set-II).

The system will generate a citation network only with the Set-I papers based on references. Following centrality measures will be used on them to determine their importance [152,153].

Motivation of Selecting Various Centrality Measures

Different centrality measures are summarized in Table 2.2. For each paper p, in-degree (p) is computed. The papers whose in-degree is greater than or equal to average in-degree of the network are shortlisted for further computation. Later on again, the average score of degree {indeg(p) + outdeg (p)} is taken into consideration for removing papers. We adopt such two-stage filtering in order to ensure that i) first, highly cited papers are not missed, and ii) no new papers which cite a lot of papers are missed either.

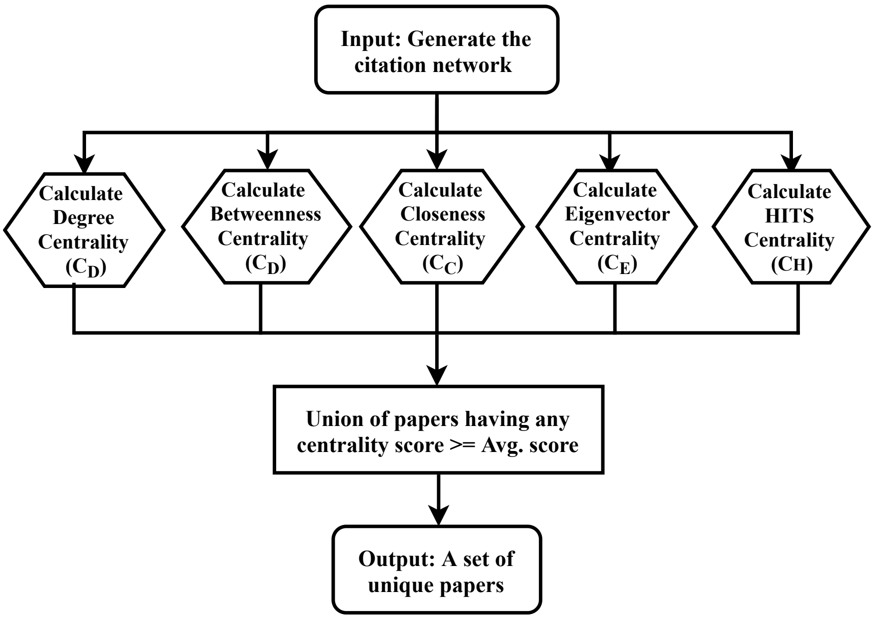

Input: Generate the citation network
Calculate Degree Centrality (CD)
Calculate Betweenness Centrality (CD)
Calculate Closeness Centrality (CC)
Calculate Eigenvector Centrality (CE)
Calculate HITS Centrality (CH)
Union of papers having any centrality score >= Avg. score
Output: A set of unique papers

Figure 3.4: Graph model for centrality measures

The average score of each measure is used as a threshold to shortlist in parallel, which are combined to filter only unique papers (Fig. 3.4). We choose all of them individually as the aim of filtering is to first remove the unimportant papers before selecting the important ones.

For example, if a very high-quality paper has low in-degree because of its recent publication, the paper may not be considered in degree centrality calculation, but it gets due consideration in Betweenness, Closeness, Eigenvector and HITS centrality calculation and, therefore, may qualify based on these measures (Fig. 3.4). This way if a paper lacks in one or more factors in the citation profile, it can qualify through other centrality measures implying fair chance to all potential papers. Moreover, this exercise is restricted only to the Set-I papers, which are having a good number of citations. The task, therefore, does not punish any papers which do not have enough citations (Set - II).

A smaller citation network is generated considering only the shortlisted papers, and the connected components (mathematically they are *weakly connected components* as directed edges will be deemed as undirected edges for finding the connected components and henceforth often referred to as merely *components*) are identified.

We remove such components having less than the average number of nodes retaining the rest where,

$$\text{Average \#nodes} = \frac{\text{Total \#nodes in citation network}}{\text{Total \#connected components}} \tag{3.1}$$

This step further reduces the number of potentially non-relevant papers.

Complexity Analysis

Although our academic bibliographic data is huge containing large number of nodes (n) (Table 3.14), the graph is actually a sparse one as the number of edges (m) is much less ($m < O(n^2)$). In this work (implementation with MAG datset), we found around 5k-13k papers (Computer Science) after keyword-based search strategy (Sec. 3.3.3). But most of the papers were there without any citations. For 72 different seed papers, the average number of papers found with citations (Set-I) were $4,373$ and average number of edges were around $7,496$. The average degree of a node was 1.71.

Algorithm 1: Similarity score generation

```
Input: $S_t$= the seed title to be compared with
       $C_t$= List of titles to be checked for similarity
Output: List of Similarity scores ($S_t, C_t$)
Function Create_Synset($S$):
    $Synset \leftarrow \{\}$
    foreach word $w \in S$ do
        $POS_w \leftarrow$ do parts of speech tagging
        $Synset \leftarrow$
        Synset $\cup$ wordnet.Synsets(w, $POS_w$)[0]
        /* adds only the first synonym for $w$ */
    end
    return $Synset$
End Function
$Synset_S \leftarrow$ Create_Synset($S_t$) /* Synset of $S_t$ */
foreach title $t_j \in C_t$ do
    $Synset_j \leftarrow$ Create_Synset($t_j$) /* $\forall t_j \in C_t$ */
    $Scores_j \leftarrow Sim(Synset_S, Synset_j)$ /* Algo. 2 */
end
return $Scores = [Scores_j]$
```

In Social network analysis, a Degree centrality measure has a time complexity of $O(m)$. Both Closeness and Betweenness centrality of all vertices in the citation network involve the shortest paths between all pairs of vertices on a graph, which takes $O(mn)$ time using Brandes algorithm [154]. This algorithm performs a simple breadth-first search, in which distance and shortest-path counts are determined from each vertex.

Algorithm 2: Synset similarity algorithm

```
Input: $Synset_S$= array synset terms $S$
       $Synset_j$ = array of synset terms $j$
Output: Similarity score ($Synset_S$, $Synset_j$)
Initialization
$Score \leftarrow 0, Word\_count \leftarrow 0$
Function $Sim(Synset_S, Synset_j)$:
    for each word $a_i$ in $Synset_S$ do
        $Best\_score \leftarrow 0$
        /* $Best\_score$ for each word in $Synset_S$ */
        for each word $b_i$ in $Synset_j$ do
            if $Wup\_sim(a_i, b_i) > Best\_score$ then
                $Best\_score \leftarrow Wup\_sim(a_i, b_i)$
                /* $Wup\_sim$ as per Eqn. 3.2*/
            end
        end
        if $Best\_score \neq 0$ then
            $Score \leftarrow Score + Best\_score$
            /* Sum of all $Best\_score$ */ $Word\_count \leftarrow Word\_count + 1$
            /* Number of words in $Synset_S$ */
        end
    end
    $Sim\_score \leftarrow \frac{Score}{Word\_count}$
    return $Sim\_score$
End Function
```

For computing the Eigenvector and HITS centrality measures, power iteration method can be used that approximates the metrics within a few steps and also avoids numerical accuracy issues. That way both Eigenvector and HITS centrality measures can be done in $O(n^2)$ time [155,156]. Hence, overall computational complexity of social network analysis is not more than $O(n^2)$.

3.3.5 B. Content Analysis

From each such shortlisted connected components, title and keywords similarity of all papers with the seed paper (input paper) are re-considered.

For title similarity, Python nltk Wordnet is utilized [157] [1], as given in Algo. 1 and Algo. 2. We employed Wu-Palmer similarity ($Wup\_similarity$) to compute the similarity among Synsets [158]. Synsets are organized in wordnet taxonomies (hypernym tree) in such a way that the root or higher-level terms are more abstract terms (hypernyms) and lower-level terms are more specific (hyponyms).

It is mainly calculated the similarity by considering the depths of two Synsets and that of their Least Common Subsumer (more specific ancestor node) [2].

$$Wup\_similarity(s_1, s_2) = 2 * \frac{depth(lcs(s_1, s_2))}{(depth(s_1) + depth(s_2))} \tag{3.2}$$

Where $0 < Wup\_similarity(s_1, s_2) \leq 1$ and lcs stands for Least Common Subsumer. The score can never be 0 as the depth of the lcs is never 0 (depth of the root of taxonomy is 1). Whenever multiple candidates for the lcs exist, the one having the longest paths to the root will be selected during the calculation.

Jaccard similarity coefficient is employed for keyword similarity (See Eqn. 3.3). If P and Q be a set of keywords extracted from seed the paper and from the test papers respectively or vice-versa [159], it is defined as

$$J(P, Q) = \frac{|P \cap Q|}{|P \cup Q|} \tag{3.3}$$

where $0 \leq J(P, Q) \leq 1$.

Later, we find the cumulative similarity score as an average of the two similarities for each paper in the connected components as follows.

$$\text{Cumulative similarity} = \frac{\text{Title similarity} + \text{Keywords similarity}}{2} \tag{3.4}$$

These cumulative similarity scores are computed for both Set-I and Set-II papers. However, the following steps are done for Set-I papers. Set-II papers join at the end of Main Path Analysis(E).

[1] It is an open-source package in Python language which is trained on English Wordnet

[2] www.nltk.org/howto/wordnet.html

Identification of Papers of Interest (I)

The cumulative similarity scores (Eqn. 3.4) are used to identify top-k (we take $k = 10$) papers from each connected component that are most similar to the seed paper. Abstracts of these top-k papers are extracted, and abstract similarity is calculated with the seed paper applying Okapi BM25+[3](Sec. 2.4.2). The paper having the highest BM25+ score (Sec) with the seed paper is chosen as the paper of interest (I) for each selected components.

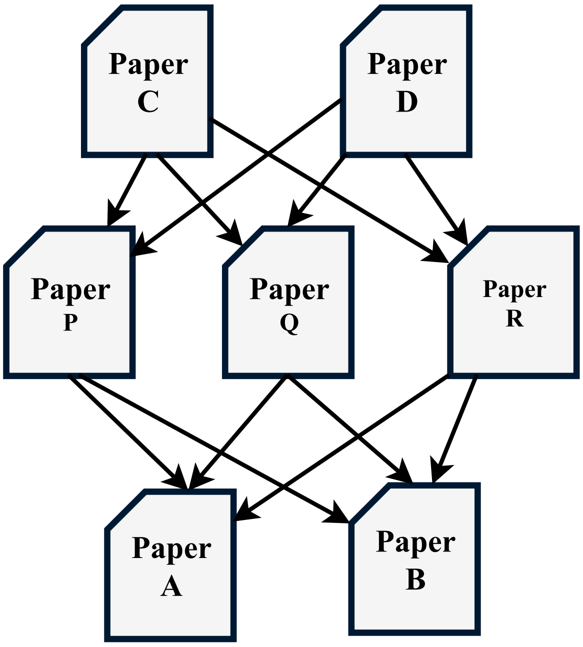


Figure 3.5: The structure of citation analysis

3.3.6 D. Citation Analysis

The papers so selected provide the basis of further study of the interplay of the papers within a component based on co-citation analysis. We look at the bibliographic coupling (BC) and co-citation (CC) scores for each candidate papers [160, 161].

[3] BM stands for Best Matching. To address the deficiency of Okapi BM25 in its term frequency (TF) normalization component, Okapi BM25+ (a variant of Okapi BM25) is employed.

Bibliographic Coupling (BC)

It gives a measure of the similarity between two papers based on the number of common papers they jointly cite.

$$\text{BC}(C, D) = |L_C \cap L_D| \tag{3.5}$$

where L_C, L_D are set of bibliographic lists in C & D respectively.

Fig. 3.5 illustrates bibliographic coupling, showing that papers P, Q and R are cited by both papers C and D. BC strength of papers C and D is, hence, 3.

Co-Citation(CC)

It denotes the number of other papers that cite two given papers together. The co-citation strength (CC-strength) can be computed as follows.

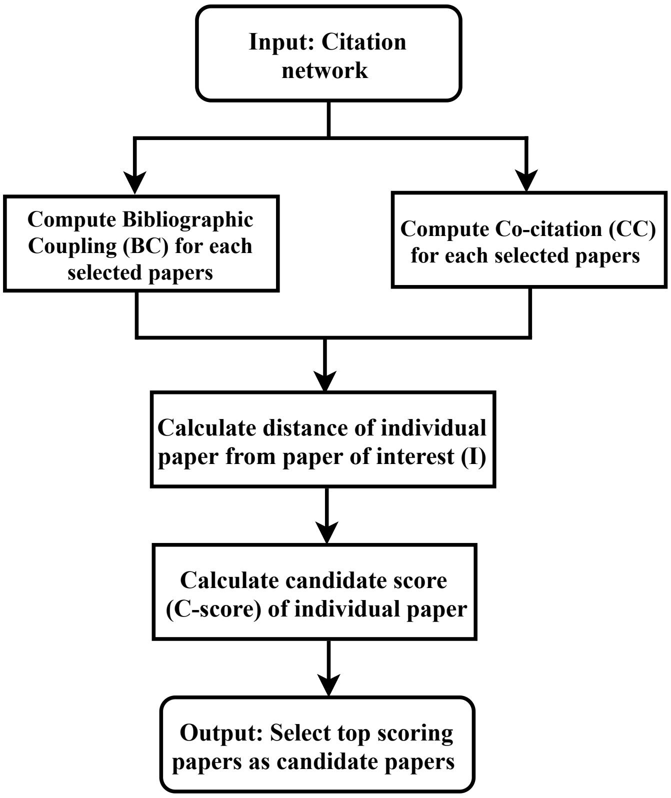

Input: Citation network
Compute Bibliographic Coupling (BC) for each selected papers
Compute Co-citation (CC) for each selected papers
Calculate distance of individual paper from paper of interest (I)
Calculate candidate score (C-score) of individual paper
Output: Select top scoring papers as candidate papers

Figure 3.6: Graph model for candidate score computation

While BC score implies similarity of two papers based on similar sub-domain and time (referring the same set of papers), CC score implies shared authority on a particular sub-domain or very general domain (same set of papers jointly refer the two). Taken together, they represent the importance and contemporariness of a pair of papers within

Table 3.1: Computation of C-score for papers in the citation network

Paper	Total BC	Total CC	Total Similarity	d(I,k)	C-score
P_6	2	6	8	2	4.0
P_{15}	0	1	1	2	0.3
P_{17}	3	1	4	2	2
P_{20}	1	2	3	3	1
P_{22}	2	6	8	3	2.6

a citation network.

$$\mathrm{CC}(A, B) = |I_A \cap I_B| \tag{3.6}$$

where, I_A denotes the set of papers that cite paper A, in other words, $|I_A| = indeg(A)$ In Fig. 3.5, $CC(A, B) = 3$ as papers A, B are co-cited by papers P, Q, and R.

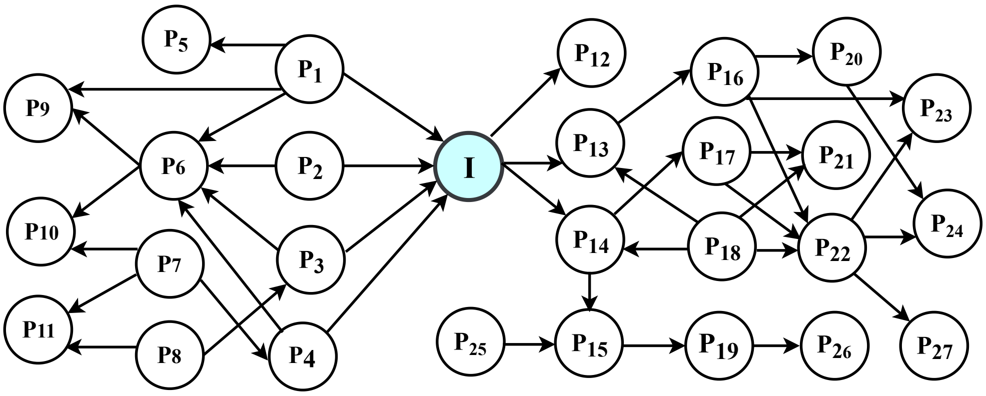


Figure 3.7: Seven-level citation network

Candidate Score Computation (C-score)

Summation of BC and CC respectively of a particular paper paired with all others in the component can, therefore, be an important feature and we combine them into a single measure called the candidate score (C-score) which is defined by

$$\text{C-score}(k) = \frac{\sum_{o \in S_I} [\mathrm{BC}(k, o) + \mathrm{CC}(k, o)]}{d(I, k)} \tag{3.7}$$

where S_I is the collection of nodes in the component with the paper of interest I, and $d(I, k)$ is hop-distance from I to k. The reason for using such hop-distance along with both BC and CC lies in the state-of-the-art literature [162].

All papers other than paper "k" are represented as "o". The C-score considers the relevance of "k" with, o and I. We normalize the score so obtained by $d(I, k)$ in order to provide incentives to the papers that are close to I and penalize the papers at a distance.

The overall process of candidate papers selection is depicted in Fig. 3.6.

Illustrative Example of Candidate Paper Selection

Fig. 3.7 shows an illustrative example with representative citation scores in Table 3.1 (Fig. 3.7 is restructured used by Son et al. [162]). P_6 has out-degree 2 meaning P_6 can have maximum two pairs with positive BC-strength. $BC(P_6, P_1) = 1 = BC(P_6, P_7)$ and BC for all other pairs of nodes involving P_6 is zero. Again, P_6 has in-degree $= 4$ meaning it can have maximum 4 pairs of positive CC-values. $CC(P_6, P_9) = 1 = CC(P_6, P_5), CC(P_6, I) = 4$. The aggregate score of the BC and CC is 8, which is the numerator of the C-score for P_6. Similarly, the denominator of the C-score for P_6 is $d(I, P_6) = 2$. Similarly, BC and CC values of all other nodes are computed. Some example nodes are given in Table 3.1.

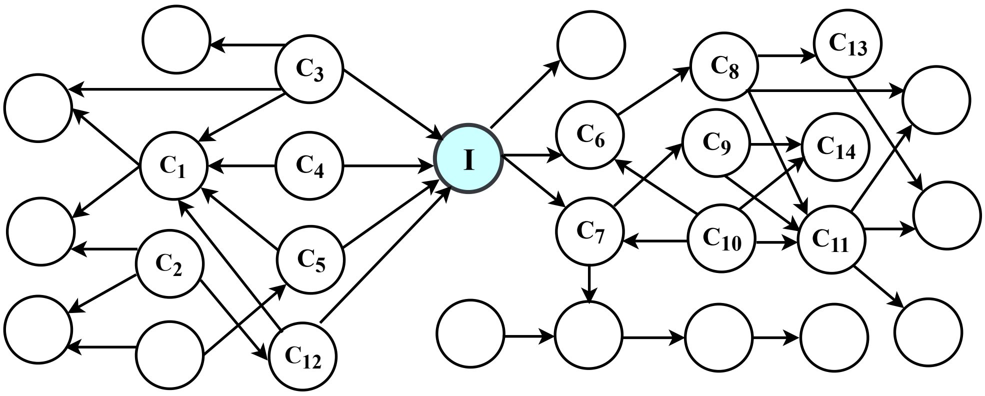


Figure 3.8: Identification of papers with higher than average C-scores

Although total similarity score (sum of BC and CC) of paper P_6 and P_{22} are equivalent, C-score of P_{22} is less than that of P_6 because P_{22} is farther from I than P_6 is. We propose that P_6 is more similar to the user topic than P_{22} is. On the other hand, although P_6 and P_{17} are at the same distance, P_{17} has a lower C-score because the total similarity of P_{17} is lower than that of P_6.

Again, BC of P_{17} is greater than BC of P_{22} but the C-score of P_{22} is higher than that of P_{17}. The papers with low C-scores tend to be isolated from the network community. P_{15} is likely to be an irrelevant paper probably produced by self-citations, and ceremonial citations[4].

Computation of C-scores are thus done for each of the papers with respect to an I in the component. Based on a user-specified threshold, the papers with higher C-scores are selected as candidate papers which are used to generate a citation network.

In this work, experiments were conducted for 72 seed papers in computer science domain and 48 seed papers in the biology domain. We observed that choosing average C-scores as the threshold fit the bill and the average number of papers left after this step was in between 800 and 2000 (Computer Science).

In the above example, initially, there are a total of 27 papers, excluding I in the citation network depicted in Fig. 3.7. Considering average C-score as a threshold as appeared in Fig. 3.8 only 14 papers are shortlisted as candidate papers.

Complexity Analysis

Let us assume there are n_1 number of vertices and m_1 number of edges exist in the citation network generated among papers shortlisted after title and keywords matching as defined in Sec. 3.3.5 (only for papers shortlisted after social network analysis). Considering the citation network contains a small diameter d, phenomenon known as "six degrees of separation" [163–165]. In this work, average number of nodes and edges were found around $1,190$ and $1,837$ respectively. The average degree of a node and average diameter were 1.54 and 8.3 respectively.

Citation analysis mainly involves two types of steps such as Bibliographic Coupling (BC), and Co-Citation (CC) in a citation network. Let's assume the maximum number outdegree, and indegree of a given vertex are k_1, and k_2 respectively. To identify the BC of a given node p_i, we need to move towards each outgoing vertices of that particular node p_i. Then for each outgoing vertices of p_i, we need to visit only those vertices whose outgoing edges are directly connected to any outgoing vertices of p_i. As a result, the total time complexity for computing BC for a single vertex is $O(k_1k_2)$. So for all the nodes in the citation network, the complexity is $O(n_1k_1k_2)$.

Similarly, for CC computation of a particular vertex p_i, we need to initially visit all the vertices whose incoming edges are directly connected to vertex p_i. Later, for each of

[4] Ceremonial citation is one that is done even though the citing paper is very lightly related with the cited publication

these vertices whose outgoing edges are connected to vertex p_i, we need to traverse all outgoing vertices. So a total of $O(k_2k_1)$ complexity is needed for one vertex. As a result, $O(n_1k_1k_2)$ complexity is needed for all the nodes in the citation network. The total time needed to compute the distance of each vertex from a given vertex p_i is $O(m_1 + n_1)$.

Hence, overall computational complexity of citation analysis is not more than $O(m_1+n_1)$.

3.3.7 E. Main Path Analysis

The citation network so produced after shortlisting of papers with C-score is used for main path analysis. Here we try to arrange the papers in a chronological fashion so that knowledge is supposed to flow from a source to sink paper. A source is an original paper that is supposed to introduce a new domain (loosely can be considered as the origin of knowledge) whereas sink is the most recent paper which cites its ancestor papers. To build this directed graph (we call it *reference-flow graph*), we need to reverse the direction of the citation network.

For example, if there is a citation from node $B \rightarrow A$ in citation network, a directed link $B \leftarrow A$ is drawn in the reference-flow graph for main path analysis. It means paper A is cited/referred by paper B. After applying the above changes in the citation network in Fig. 3.8, the new reference-flow graph is obtained as given in Fig. 3.9.

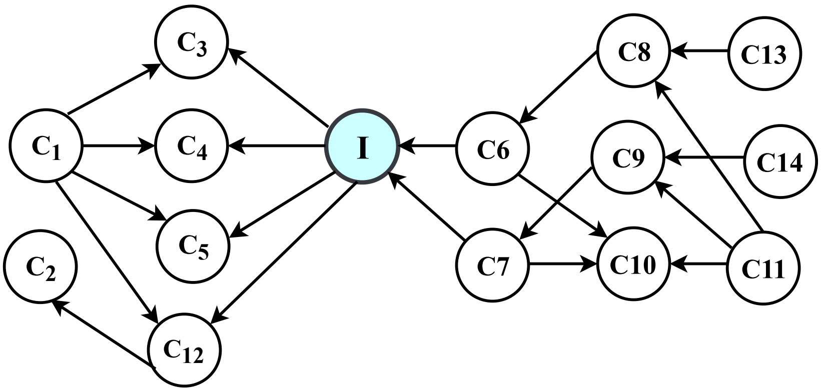


Figure 3.9: Selection of candidate papers

The most significant path in a citation network is traced by main path analysis. This is used to identify the structural backbone in the evolution of a scientific field [166]. Main path analysis is more useful when there is a need to investigate connectedness in acyclic networks and especially draw attention when nodes are time-dependent, as it chooses the most representative nodes at different points of time [167].

To determine the main path, the following steps are needed to be considered.

(i) Assignment of link-weights in the network using traversal count.

(ii) Identification of key route in reference-flow graph.

Assignment of Link Weights in Citation Network

For identifying the main path in any network, the links in the network are assigned weights using traversal count [168] that measures the importance of a link. The number of traversals of a link for different source-sink pair of nodes is known as traversal count of the link. It has several variants, depending on how the pairs are chosen. We are using Search Path Count (SPC) [5] for weighing the links.

If a path through which much knowledge flows includes a citation link, it has a certain prominence in the knowledge-dissemination process. Using SPC as traversal count in Fig. 3.9, we obtain the link- weight of the citation network as given in Fig. 3.10. The most significant links are added to form the main path connecting a source and a sink node [167] as described below.

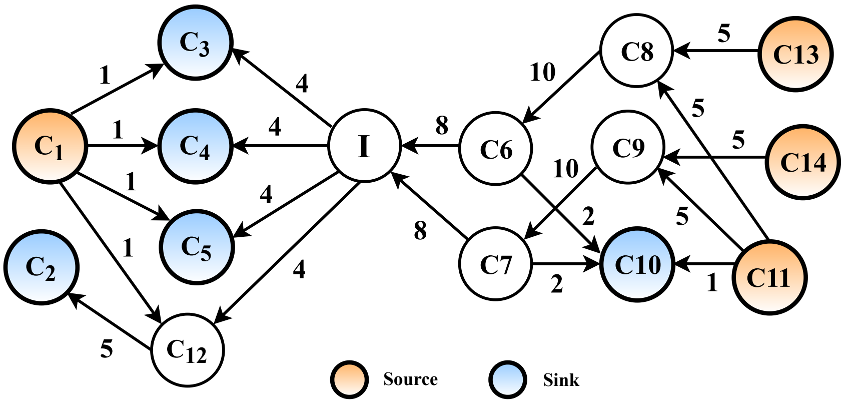


Figure 3.10: Assignment of link-weights using SPC technique

[5] A link's SPC is the number of times the link is traversed if one runs through all possible paths from all the sources to all the sinks.

Identification of Key-route in Citation Network

There can be several paths between a source and sink pair having the same traversal count. We need to select the most promising one. Local, global, and key-route search are some of the various approaches to identify it. We find the key-route search in the following way.

(i) Choose the link having the maximal traversal count. This link is considered starting link of key-route.

(ii) Push ahead from the end node of the key-route until the point when a sink node happens.

(iii) Go in reverse from the begin node of the key-route until the point that a source node happens.

By executing the steps many times, multiple key-routes can be found, each time choosing the link with the next-highest traversal count. However, the first such key-route with highest SPC is considered as the *main path.*

Illustrative Example of Key-route Identification

Let us consider a simple citation network depicted in Fig. 3.9. It has 4 sources, C_1, C_{11}, C_{13}, C_{14} and 5 sinks such as C_2, C_3, C_4, C_5, C_{10}. There are various substitute paths to traverse from the sources to the sinks. Assuming that one exhaustively searches all paths from every source to every sink, the SPC for each link is defined as the total number of times the link is traversed. For example, Link $C_{12} - C_2$ has an SPC value of 5 because paths $C_{13} - C_8 - C_6 - I - C_{12} - C_2$, $C_{14} - C_9 - C_7 - I - C_{12} - C_2$, $C_1 - C_{12} - C_2$, $C_{11} - C_8 - C_6 - I - C_{12} - C_2$ and $C_{11} - C_9 - C_7 - I - C_{12} - C_2$ pass through it. Since the Link I-C_5 is a part of four distinct paths $C_{11} - C_8 - C_6 - I - C_5$, $C_9 - C_7 - I - C_5$, $C_{13} - C_8 - C_6 - I - C_5$, and $C_{14} - C_9 - C_7 - I - C_5$ respectively, its SPC value is 4. The link-weights as SPC are shown in Fig. 3.10.

Links $C_9 - C_7$ and $C_8 - C_6$ have the highest SPC value of 10. Initially, the link $C_9 - C_7$ and $C_8 - C_6$ are chosen due to their highest SPC value as 10. Now, search backward from the beginning node until a point that source is hit and search forward from the end nodes C_7 and C_6 until a sink is hit.

We get $C_{11} - C_9 - C_7 - I - C_{12} - C_2$, $C_{11} - C_8 - C_6 - I - C_{12} - C_2$ as the global key-routes starting from node C_{11}. In addition, we get $C_{13} - C_8 - C_6 - I - C_{12} - C_2$, $C_{14} - C_9 - C_7 - I - C_{12} - C_2$ as the global key-routes. The sum of the SPC values in all the key-route paths is 32, which, is the largest among all possible paths, as shown in Fig 3.11. We have a total of 10 papers in the key-routes, including the paper of interest I. After retaining only unique papers in the key-route, we have C_2, C_6, C_7 ,C_8, C_9, C_{11}, C_{12}, C_{13}, C_{14} and I as the final candidate papers. Initially, we had a total of 28 papers, including I in the network. After applying $C - score$, there were 15 papers, including I that were shortlisted. Now, after applying the key-route, we end up with 10 papers, including I as significant papers in the citation network (Fig. 3.11).

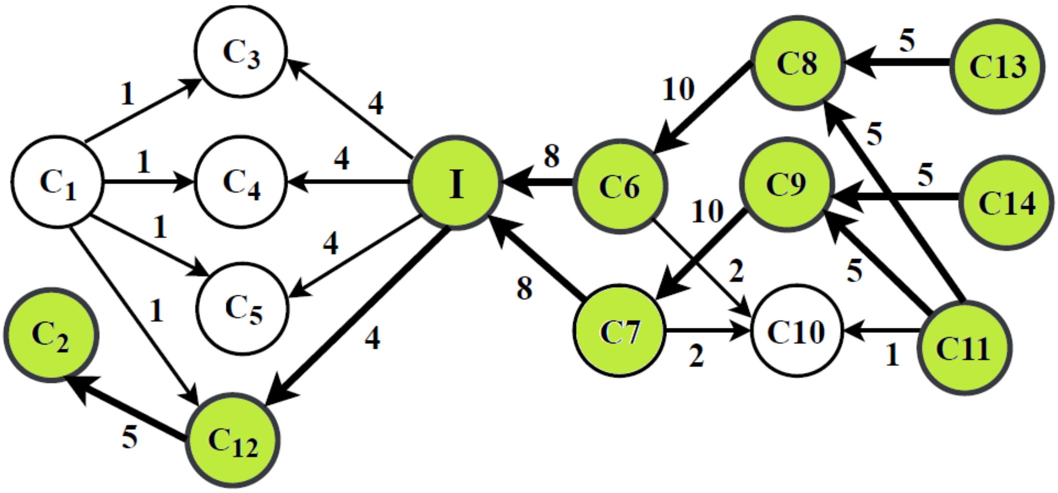


Figure 3.11: Key-route identification using main path analysis

Complexity Analysis

In the main path analysis, we need to traverse all the outgoing vertices of a given source vertex s_i. Then repeatedly for each outgoing vertices of s_i we need to traverse their corresponding outgoing vertices till we reach a sink node s_j in the citation network. In this work, average number of nodes and edges were found around 503 and 872 respectively with an average degree of 1.73. It is clearly indicates the sparseness of the citation network. The diameter of the network was 6.4. The maximum outgoing vetices and maximum incoming vertices of a node were around 23 and 57 respectively. We need to visit all possible paths from a given source vertex to a sink vertex.

As mentioned in Sec. 3.3.6, the diameter of the citation network is d, and the number of outdegree of a given vertex is k_3. So for a given source vertex s_i, the time complexity will be $O(k_3^{d-1})$. Assume that there are s_1 number of source vertices and s_2 number of sink vertices present in the citation graph. So the total time complexity will be $O(s_1 k_3^{d-1} s_2)$.

Hence, overall computational complexity of main path analysis is not more than $O(s_1 k_3^{d-1} s_2)$.

3.3.8 F. Result Fusion

To find the final ranking of venues, the following steps are followed sequentially.

(i) Merging Set-I and Set-II papers for abstract similarity using LDA and NMF techniques

(ii) Extraction of unique venues and their similarity computation

(iii) Normalization of similarity scores and their fusion

Merging Set-I and Set-II Papers for Abstract Similarity Using LDA and NMF Techniques

Title similarity and keyword similarity are computed for Set-II papers as well using Algo. 1 and Algo. 2 and keyword matching is performed by Eqn. 3.3. Top t_2 similar papers are chosen based on cumulative scores.

We have three assumptions regarding the inclusion of Set-II papers for abstract similarity.

(a) There may be few papers that have no citations (Set-II), but any such paper may be published at reputed venues.

(b) The title and keywords of the seed paper may match with some papers in Set-II, so there is a possibility that the seed paper may get accepted at similar venues as that of Set-II papers.

(c) Generally, the papers published in reputed venues get a high number of citations.

So along with the shortlisted key-route papers (t_1), we add shortlisted Set-II papers (t_2) to make a combined list of shortlisted papers ($t_1 + t_2$).

For each paper in the list, we find abstract similarity with the seed paper using LDA and NMF techniques independently. We use LDA and NMF as these techniques capture topics rather than exact terms. At this point, when sufficient care has been taken on keywords match and also some papers are qualified based on other criteria (not keyword matching), we believe some abstract level topic matching would work.

Also, we use both LDA and NMF separately since LDA mainly considers terms in a document independent of its presence in other documents to identify topics [132]. NMF, on the other hand, tries to capture a set of words occurring together in a topic using tf-idf vector [46]. We assume these two methods are complementary in nature and provide two different ranks to a given paper. The number of topics or vector dimension considered here is 100 during the topic extraction. Thus we get two lists of papers sorted in decreasing order of similarity scores based on LDA and NMF respectively.

Extraction of Unique Venues and Their Similarity Computation

We extract the venues corresponding to the papers in the two ranked lists and make two lists of unique venues. Venues are ordered based on the top-scoring papers published there. We also assign the score to the venues such that each venue v_k represents the highest similarity score of papers published at v_k. Hence we have two ordered lists of venues. The entries in the lists are the same set of venues occurring possibly at different ranks with different similarity scores.

Normalization of Similarity Scores and Their Fusion

The similarity scores depend on the techniques (LDA or NMF) and hence are not readily comparable. To compare, we need to normalize the scores and then apply some fusion techniques to get a single ordered list of venues. In the recommendation community, data fusion is one of the widely investigated areas. We used score-based fusion over rank-based one as it reduces the number of ties. There are several score-based fusion techniques such as CombSum, CombMNZ, and weight combination [169].

We use *CombMNZ* fusion technique [170]. To run CombMNZ, similarity scores of two lists must be normalized, so that they lie in a common range. There are different normalization strategies proposed in the literature. We select the one used by Lee et al. [171], as it is the one most commonly used for comparison and has been defined as

```
Algorithm 3: Fusion-based final venues ranking
Input: Observed shortlisted key-route papers ($t_1$) and shortlisted Set-II papers ($t_2$)
       to check abstract similarity
Output: Top N recommended list of venues
Initialization
let m be the seed paper
let $T = t_1 + t_2$ be the set of candidate papers
while $T$ is not empty do
    for $i \leftarrow 0$ to $|T - 1|$ do
        Compute abstract similarity with m using LDA technique
        $L_i \leftarrow$ similarity score of each candidate papers
        Compute abstract similarity with m using NMF technique
        $N_i \leftarrow$ similarity score of each candidate papers
    end
    $S_l$=Ordered list of papers in decreasing values of LDA based similarity score
    $V_l$=Set of unique vanues based on top scoring papers in $S_l$
    for $k \leftarrow 0$ to $|V_l - 1|$ do
        $l_k \leftarrow$ LDA based similarity score of each candidate venues
        $n_k \leftarrow$ NMF based similarity score of each candidate venues
    end
    Normalize($l_k$) $\leftarrow \frac{l_k - min(l_k)}{max(l_k) - min(l_k)}$;
    Normalize($n_k$) $\leftarrow \frac{n_k - min(n_k)}{max(n_k) - min(n_k)}$;
end
for $k \leftarrow 0$ to $|V_l - 1|$ do
    N($v_k$)=Normalize($l_k$) $+ Normalize(n_k)$
    $CombMNZ_{v_k} = N(v_k) * |N_s > 0|$
end
N($v_k$) $\leftarrow$ Normalized score of venue $v_k$ in result set LDA and NMF
$|N_s| \leftarrow$ No. of non-zero normalized scores given to $v_k$ by any result set LDA and NMF
Sort venues in the decreasing order of $CombMNZ_{v_k}$ scores
Prepare the final list of top N venues recommendation
```

"standard normalization" [172]. We apply $CombMNZ_v$ on normalized scores of venues. Finally, we recommend top-N venues based on $CombMNZ_v$ scores where N (usually $N \neq t_1$ or t_2) is user-specified. The complete algorithm is provided in Algo. 3.

3.4 Experiments

We conduct an extensive set of experiments. Below we outline the experimental dataset, evaluation strategy, evaluation metrics, experimental setting, parameter tuning, and other comparable methods. All experiments are conducted on a 64-bit and 2.4GHz Intel Core i5, 8-GB memory system.

3.4.1 Dataset Used

We use the Microsoft Academic Graph (MAG) dataset [44,147] (Sec. 2.7.1) to demonstrate the effectiveness of DISCOVER.

3.4.2 Evaluation Strategy

We adopt two kinds of evaluations namely Coarse-level or offline evaluation and Fine-level or online evaluation to measure the performances of DISCOVER against other state-of-the-art methods (Sec. 2.5).

3.4.3 Evaluation Metrics

We employ eight metrics such as Accuracy@N, Mean Reciprocal Rank (MRR), Precision, $F{-}measure_{macro}$ (F_1), Normalized discounted cumulative gain (nDCG), Diversity, Stability, and Average-Venue Quality (Ave-quality) to evaluate the performance of DISCOVER against other state-of-the-art methods (Sec. 2.6).

3.4.4 Experimental Setting

Initially, we consider papers from CS. Removing the papers with no venues details, there are only 13,402,547 papers in CS. Similarly, we preprocess papers from BIO domain and are left with only 12,848,227 papers. All the papers published on or after the year 1982 and before the year 2012 are used as a training set, the rest (papers dated in or after 2012) are as the testing set as given in Table 3.2.

Table 3.2: Statistics of training and testing dataset in CS and BIO

FOS	Pre-processed Dataset	Training Dataset	Testing Dataset
Computer Science (CS)	13,402,547	10,424,960	2,977,587
Biology (BIO)	12,848,227	9,961,893	2,886,334

Preparation of Test Dataset

Due to operational constraints, only 12 sub-domains of CS and 6 sub-domains of BIO are selected as in testing dataset. A total of 72 seed papers are chosen from 12 sub-domains (6 from each): Information Retrieval (IR), Image Processing (IP), Security (SC), Wireless Sensor Network (WSN), Machine Learning (ML), Software Engineering (SE), Computer Vision (CV), Artificial Intelligence (AI), Data Mining (DM), Natural Language Processing (NLP), Parallel and Distributed Systems (PDS) and Multimedia (MM) in CS. Similarly, a total of 48 seed papers are chosen from 6 sub-domains (8 from each sub-domains): Computational Biology (CB), Anatomy (AN), Immunology (IM), Toxicology (TX), Biochemistry (BI) and Paleontology (PL) in BIO.

Seed papers are chosen, keeping in mind the cold-start issues for new venues and new researchers. We consider 3 categories of venues and 3 categories of researchers based on venue count (v_c) (number of papers published at a given venue) and publication count (p_c) (the number of publications of a researcher) [31, 74] on the following six categories.

(i) Category 1 : $2 \leq v_c < 8$

(ii) Category 2 : $8 \leq v_c < 15$

(iii) Category 3 : $15 \leq v_c$

(iv) Category 4 : $2 \leq p_c < 8$

(v) Category 5 : $8 \leq p_c < 15$

(vi) Category 6 : $15 \leq p_c$

It is ensured that each category is well represented in the seed papers for both CS and BIO data.

Procedure of Online Evaluation

For this evaluation, we did not have the ready annotation, but we need one. The annotation or relevance assessment is collected from the volunteers through crowdsourcing in the best effort basis. For CS, 40 researchers with expertise in the mentioned sub-domains are provided with input and output of our recommender system where for each paper, 15 venues are recommended. Out of 40 researchers, 9 evaluated 3 papers each, 14 researchers evaluated 2 each, and the rest 17 were evaluated by 17 researchers. Similarly, for BIO, 25 researchers volunteered. There were 7 researchers who evaluated 3 papers each, 9 researchers evaluated 2 each, and the rest 9 researchers evaluated one each.

All the experts were identified from academia with a minimum of 3 years of research experience. Most were having a Ph.D. except few research students and research assistants who were pursuing Ph.D with bachelors' or masters' degree in science or technology. The experts or researchers were so chosen that their active areas of research perfectly match with the topics of seed papers. Among 65 researchers, there were 12 professors, 9 associate professors, 24 assistant professors, 13 senior research students, and the remaining 7 were research assistants.

All experts were from reputed institutions like Indian Institute of Technology Kharagpur, Indian Institute of Technology Roorkee, Indian Institute of Technology (BHU) Varanasi, Central University Hyderabad, Manipal University, and Banaras Hindu University (BHU). The age range of all professors are in the range of [48-55], age range of associate professors are in between [43-47], assistant professors are having an age of [36-41], senior research students are in the age range of [28-31], and remaining research assistants are having an age range of [29-33]. The overall gender distribution of male and female experts were 44 and 21, respectively.

The experts check the titles, abstracts, authors, year of publication, and recommended venues of the papers. An expert assigns an appropriate relevance value (r) to each recommended venue as she deems the quality of the match between the scope of the recommended venue and the topic of the seed paper as below.

$$\text{Relevance } (r) = \begin{cases} 2 & \text{perfectly matching} \\ 1 & \text{partial matching} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

However, as precision is defined for binary relevance only, during precision score computation, relevance grade 2 is only considered relevant, and both relevance grade 1 and 0 non-relevant.

3.4.5 Parameter Tuning and Optimization

DISCOVER has a few essential parameters during its process pipeline as follows.

(i) Number of top-k papers for identifying I (Refer Sec. 3.3.5)

(ii) Number of papers without citation history (t_2) (Refer Fig. 3.2 and Sec. 3.3.4)

(iii) Vector dimension for LDA (Refer Sec. 3.3.8)

Impact of Top-k Papers on Selection of I

To identify the paper of interest (I), one for each component, the number of papers extracted after finding the title and keyword similarity is an important parameter to our system. Initially, we test with top 5 papers based on the cumulative score (of title and keywords matching). We then test with 10, 15, 20, 25 and 50 respectively and observe that there were not much changes on the selection of I after top-10 papers. Hence, $k = 10$ is considered for abstract similarity.

Impact of t_2 on Reccommendation Order

We also experimentally test the effect of the number of papers (t_2) selected from Set-II to perform abstract similarity. We changed the value of t_2 from 5 to 50. The upper limit is taken as 50 to offer equal opportunity to Set-II as given to Set-I (on an average the main path analysis results in 45-85 number of papers). However, it is noticed that after 15 papers, there was no major change in the recommended order, and hence, t_2 is set to 15.

Impact of Vector Dimension on Final Recommendation

In order to find the appropriate value of dimension (no. of topics) for LDA, we tried with the values$\{10, 50, 100, 200\}$. It is observed that the model performs best when the value of the vector dimension is 100. While considering the vector dimension as 200 it performs the second-best and shows the worst performance at vector dimension 10. Although 100 may not be ‘the best’ value for dimension, at the coarse level, this is the optimized number of dimensions.

To comprehensively evaluate our proposed method and more specifically, to address the broad research questions (RQs) discussed in Sec. 1.5, we prefer to examine the following sub-queries (SQs):

SQ1: How effective is DISCOVER in comparison to other state-of-the-art methods and other freely available online services?

SQ2: How is the quality of venues recommended by DISCOVER in comparison to other state-of-the-art methods in terms of H5-index of recommended venues?

SQ3: How does DISCOVER handle cold-start issues for new researchers and new venues and also the issues like data sparsity, scalability, diversity, and stability?

3.4.6 Baseline Methods

Performance of DISCOVER is compared with the eight state-of-the-art methods (Sec. 2.8.1).

Comparison with Other Freely Available On-line Services

(a) EJF: The system uses NLP and Okapi BM25 to recommend journals based on title and abstract of the seed paper [50].

(b) SJS: It is also a freely available online service which could provide journals recommendation based on the input title, abstract and field of study of the seed paper.

We also compare our results against EJF and SJS in terms of metrics discussed in Sec. 2.6. For comparison, recommendations from DISCOVER are restricted to only Elsevier and Springer journals.

3.5 Results and Discussions

The performance of DISCOVER against the existing state of the art methods and freely available on-line services EJF and SJS respectively are reported. For clarity and easy understanding, we provide the results and discussion in two steps as given below. We also

Table 3.3: Accuracy@k and MRR results comparison of FOS "CS"

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
FB	0.013	0.027	0.069	0.097	0.180	0.022
CF	0.027	0.041	0.097	0.138	0.208	0.025
CN	0.027	0.069	0.097	0.152	0.236	0.029
CBF	0.041	0.097	0.166	0.222	0.291	0.038
CF+CBF	0.053	0.102	0.179	0.237	0.319	0.047
RWR	0.055	0.111	0.180	0.236	0.347	0.058
PVR	0.083	0.125	0.208	0.277	0.388	0.062
PAVE	0.125^{+}	0.194^{+}	0.250^{+}	0.291^{+}	0.416^{+}	0.093^{+}
DISCOVER	0.222*	0.347*	0.472*	0.541*	0.708*	0.167*

'*' denote statistically significant results over the second best ('+')

conduct paired-samples t-test on overall precision, nDCG, Accuracy, and MRR for both CS and BIO between DISCOVER and the second-best performers. Only p values less than 0.05 were considered statistically significant at 5% level of significance (α =0.05).

During the assessment, stistically significant results and the second-best performer are marked by the '*' and '+' symbol in each position.

3.5.1 Offline or Coarse-level Evaluation

Venue-prediction accuracy of DISCOVER is measured on both CS and BIO domains at different recommended ranks (@3, @6, @9, @12, and @15) in Table 3.3 and Table 3.4 respectively. Prediction accuracy of DISCOVER is the best among all at all levels in the domain of both CS and BIO. Also, the scores are statistically significant from the second-best scores.

Our approach is not biased either in favor of CS or against BIO. Both the collections are also comparable (CS has 15,641,658 papers, while BIO 14,785,486 papers). But at early positions, the system performs inferior for BIO due to possibly the following reason. We observe that in BIO domain higher number of papers are shortlisted after abstract similarity calculation (70-110 in comparison to 60-100 in CS) leading to the higher number of journals in the candidate set of recommendations (Table 3.14). BIO journals are found to have larger scope covering diverse topics of papers. Hence it becomes difficult for DISCOVER to correctly predict the original journal at early ranks as a lot of BIO journals share overlapping scopes. The phenomenon is substantially less prominent in CS dataset,

Table 3.4: Accuracy@k and MRR results comparison of FOS "BIO"

Approach	Acc@3	Acc@6	Acc@9	Acc@12	Acc@15	MRR
FB	0.000	0.020	0.062	0.104	0.187	0.021
CF	0.000	0.044	0.104	0.166	0.229	0.028
CN	0.020	0.062	0.125	0.187	0.250	0.030
CBF	0.020	0.083	0.125	0.187	0.270	0.032
CF+CBF	0.032	0.097	0.136	0.203	0.291	0.044
RWR	0.041	0.083	0.145	0.229	0.312	0.036
PVR	0.041	0.104	0.187	0.250	0.354	0.039
PAVE	0.056^{+}	0.145^{+}	0.229^{+}	0.354^{+}	0.437^{+}	0.067^{+}
DISCOVER	0.128*	0.223*	0.328*	0.491*	0.697*	0.163*

'*' denote statistically significant results over the second best ('+')

possibly leading to better performance there.

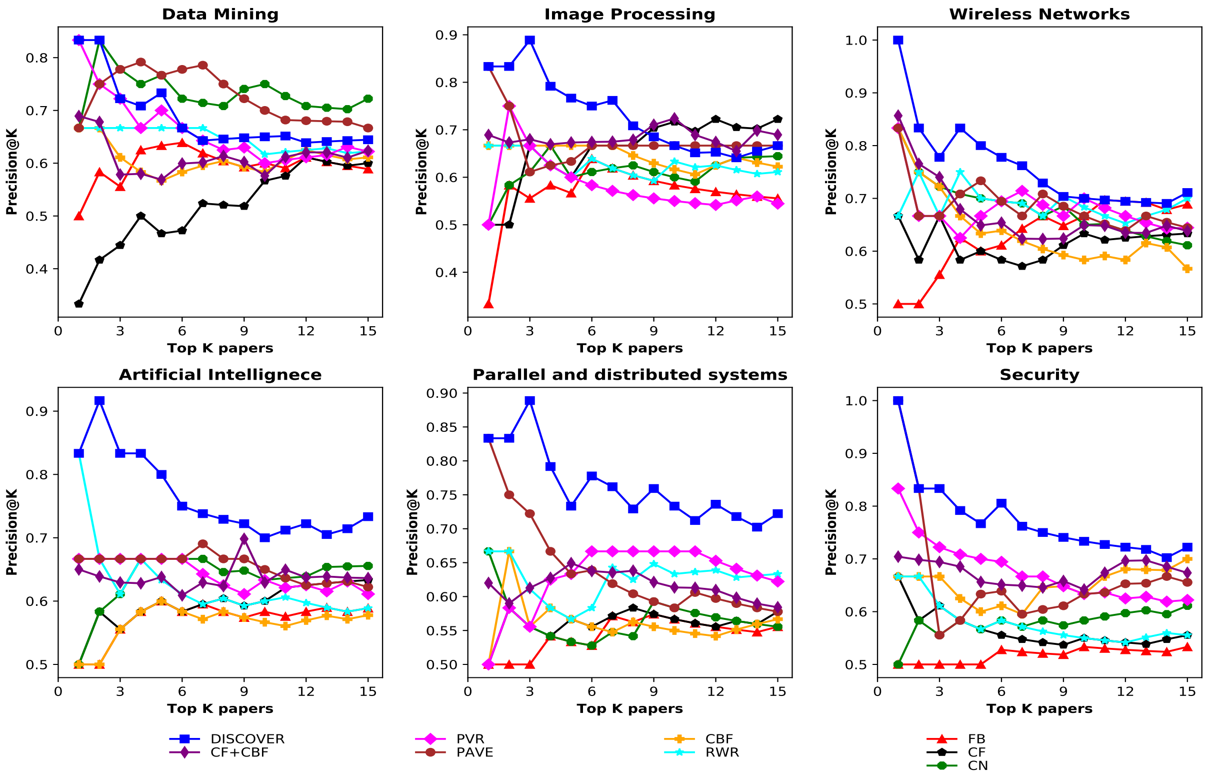


Figure 3.12: Sub-domain wise precision@k calculation (CS)

FB and CF methods exhibit the worst performances and are unable to predict at all at the 3-recommendations level. As far as MRR scores are concerned, DISCOVER displays the best, and in case of BIO, the score is more than double of its nearest competitor.

We have also investigated the efficacy of the proposed model DISCOVER in terms

Table 3.5: F-measure (F_1) analysis of FOS "CS"

Approach	F_1@3	F_1@6	F_1@9	F_1@12	F_1@15
FB	0.004	0.021	0.032	0.029	0.024
CF	0.007	0.029	0.037	0.034	0.031
CN	0.010	0.041	0.068	0.056	0.052
CBF	0.013	0.049	0.083	0.079	0.075
CF+CBF	0.016	0.053	0.098	0.093	0.088
RWR	0.018	0.058	0.103	0.109	0.102
PVR	0.023	0.064	0.194	0.176	0.153
PAVE	0.059^+	0.119^+	0.243^+	0.213^+	0.196^+
DISCOVER	0.147*	0.197*	0.363*	0.341*	0.335*

'*' denote statistically significant results over the second best ('+')

Table 3.6: F-measure (F_1) analysis of FOS "BIO"

Approach	F_1@3	F_1@6	F_1@9	F_1@12	F_1@15
FB	0.000	0.018	0.031	0.030	0.027
CF	0.000	0.026	0.039	0.038	0.034
CN	0.007	0.038	0.071	0.063	0.058
CBF	0.009	0.041	0.075	0.068	0.067
CF+CBF	0.012	0.046	0.078	0.064	0.059
RWR	0.014	0.048	0.097	0.093	0.091
PVR	0.018	0.059	0.183	0.168	0.147
PAVE	0.053^+	0.108^+	0.237^+	0.218^+	0.193^+
DISCOVER	0.119*	0.186*	0.327*	0.314*	0.308*

'*' denote statistically significant results over the second best ('+')

of $F - measure_{macro}$ (F_1) on both CS and BIO domains, as defined in Eqn. 2.25. Note that here, precision is considered only for the original venues, i.e., non-zero precision comes only if a system within top-15 recommendations recommends the original venue. In both CS and BIO domains, DISCOVER outperforms other state-of-the-art methods at all ranks (Table 3.5 and Table 3.6). Similarly, the second-best performance is exhibited by PAVE, whereas FB performs the worst in both CS and BIO. F_1 scores are generally seen to increase with rank up to a certain point (around 9-12) and drop after that. This is possible since precision and recall both increase till that point until the original venues are retrieved, causing an increase in the F_1 score. However, with further increase in ranks, precision drops sharply without much increase in recall leading to an overall drop in F_1 scores. Here also DISCOVER outperforms in terms of F_1 measure in comparison to other state-of-the-art methods.

3.5.2 Online or Finer-level Evaluation

The performances of different systems along with DISCOVER in individual sub-domains of CS (12 sub-domains) and BIO (6 sub-domains) according to different metrics are discussed below.

Precision@k

The precision scores of DISCOVER and other state-of-the-art methods of 12 sub-domains under FOS CS are shown in Fig. 3.12 and Fig. 3.13 and for 6 sub-domains of BIO in Fig. 3.14.

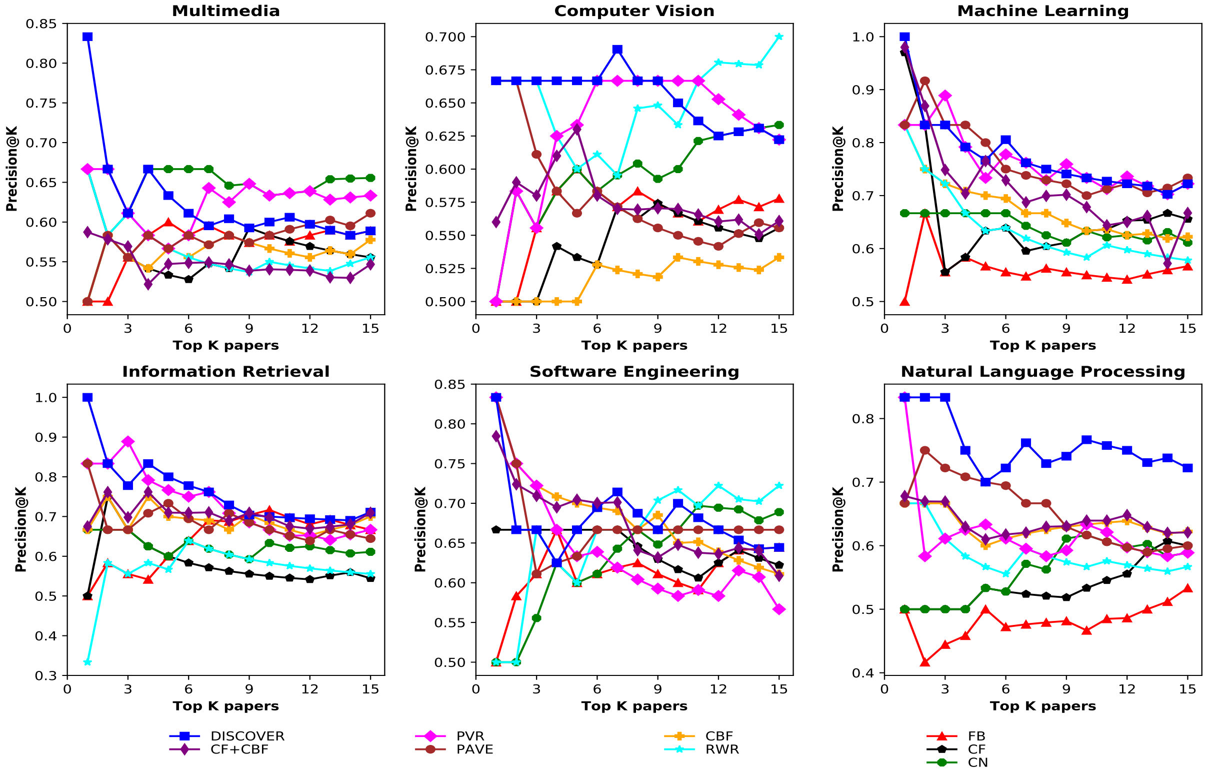


Figure 3.13: Sub-domain wise precision@k calculation (CS)

DISCOVER outperforms other state-of-the-art methods consistently according to precision@k in 9 sub-domains, namely, CV, ML, IR, NLP, IP, WSN, AI, PDS, and SC. It exhibits an average performance in domains SE, MM, and DM. In the case of BIO, DISCOVER outperforms others in all 6 sub-domains: CB, AN, IM, TX, BI, and PL.

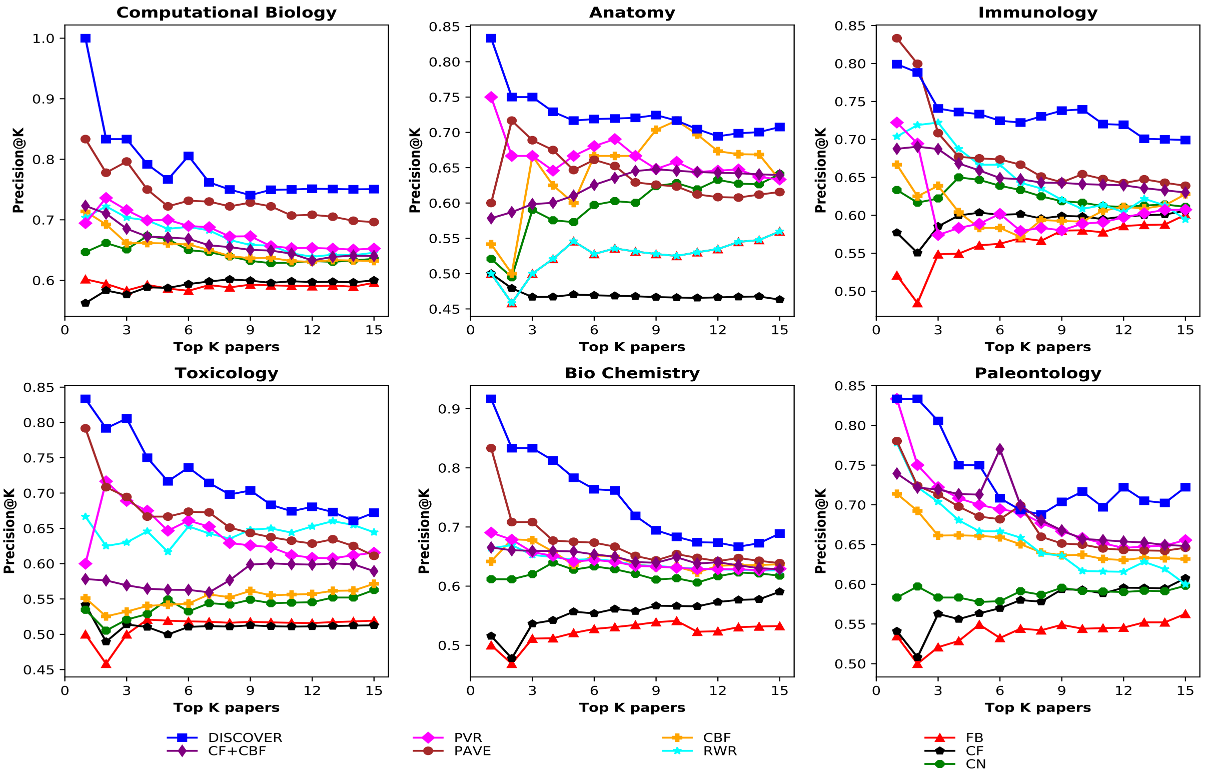


Figure 3.14: Sub-domain wise precision@k calculation (BIO)

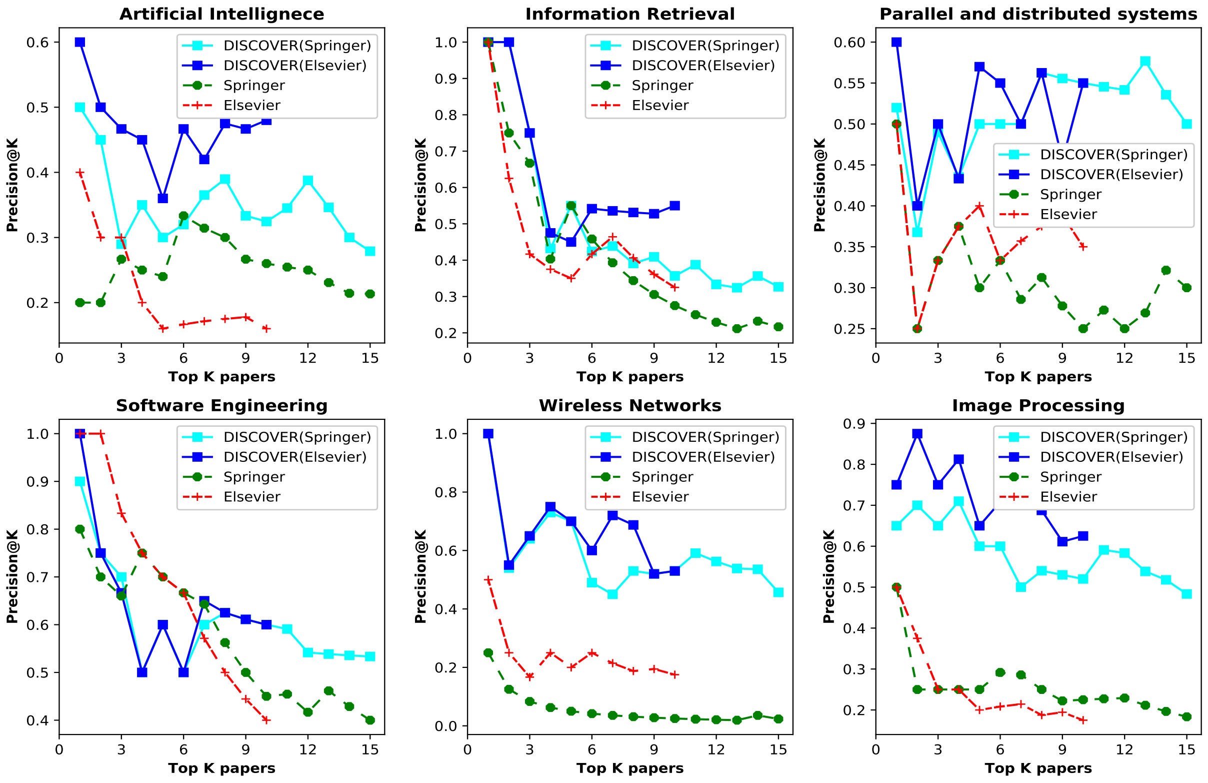
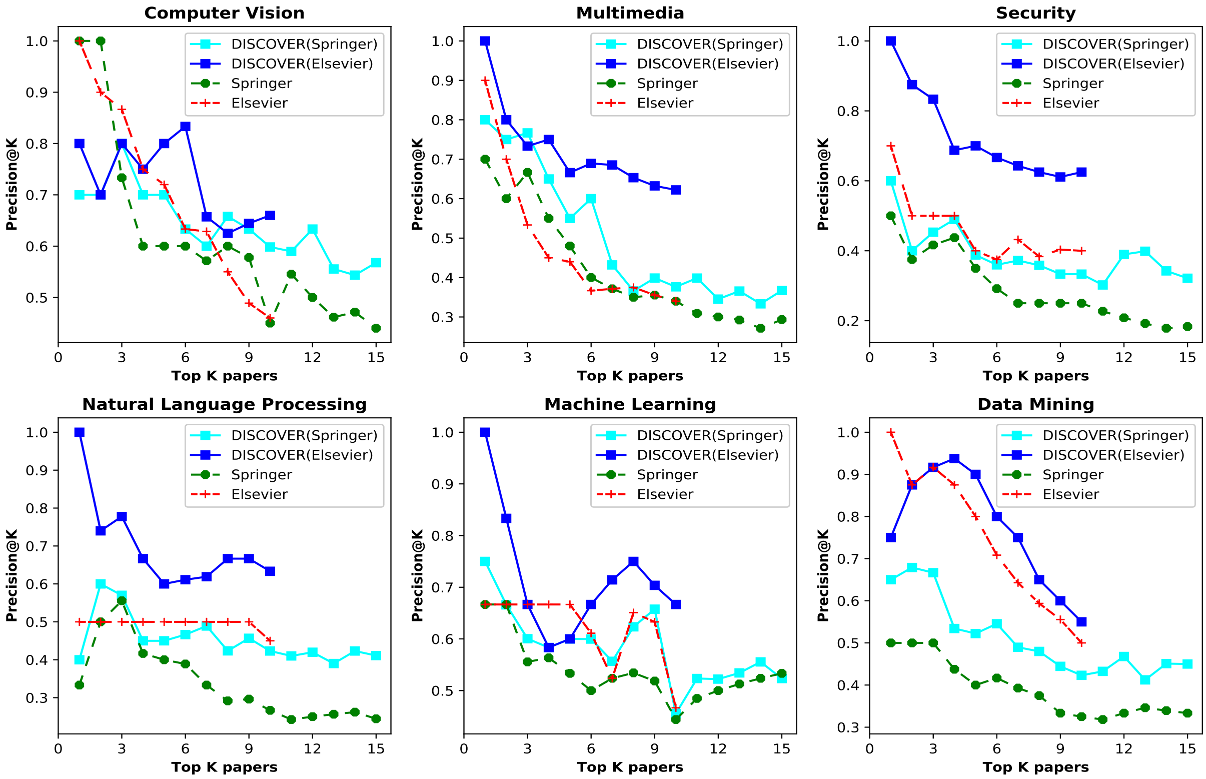


Figure 3.15: Sub-domain wise precision@k calculation (CS)

To compare against freely available online services such as EJF and SJS, recommendations of DISCOVER are restricted to only the Elsevier and Springer journals (See Fig. 3.15 and Fig. 3.16). In CS, precision@k scores of DISCOVER (Elsevier) exceed that of EJF in 10 sub-domains (AI, IR, PDS, WSN, IP, MM, SC, NLP, ML, and DM) except in SE and CV sub-domains where DISCOVER could not show good performances.



Computer Vision

Multimedia

Security

Natural Language Processing

Machine Learning

Data Mining

Figure 3.16: Sub-domain wise precision@k calculation (CS)

Similarly DISCOVER (Springer) exceeds SJS in 10 sub-domains (AI, PDS, WSN, IP, CV, MM, SC, NLP, ML, and DM) except IR and SE (Fig. 3.15 and Fig. 3.16).

In the case of BIO, DISCOVER (Elsevier) exceeds EJF in 5 sub-domains (IM, AN, TX, BI, and PL) other than domain CB (Fig. 3.17). However, DISCOVER (Springer) outperforms SJS in all sub-domains of BIO. Interestingly, DISCOVER (Springer) exhibits the best performance in CB there.

Overall Results of Precision@k

When we compute the overall precision taking the average of precision values over 12 sub-domains of CS at a given rank (3, 6, 9, 12 and 15), DISCOVER outshines other methods at all ranks (Table 3.7). PAVE is the second-best performer and mostly outplays other

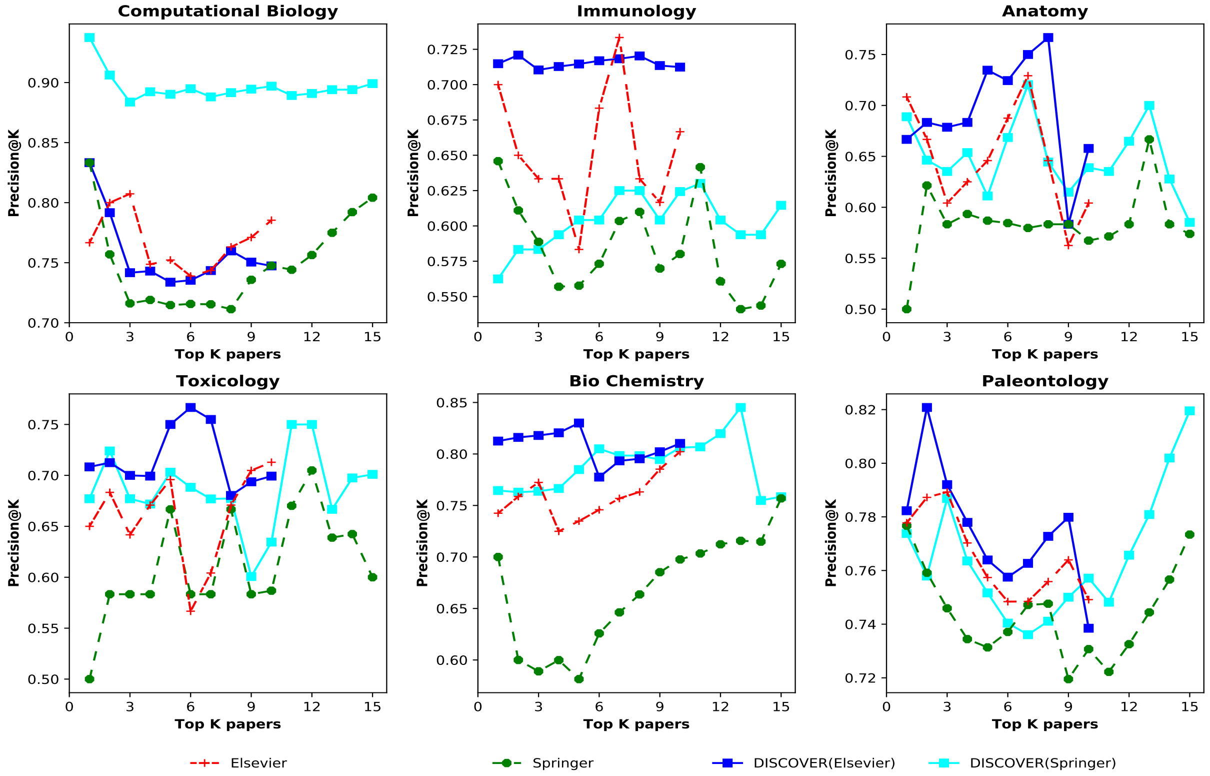


Figure 3.17: Sub-domain wise precision@k calculation (BIO)

baseline methods. As the second-best, PVR outperforms PAVE only for precision@3. Among the low-performers, CN fares badly at position (P@3) but does better than CBF at all other positions. FB method performs the worst compared to all other methods except at position 6 (P@6).

For BIO, the same exercise was done (Table 3.8) with similar results. DISCOVER is consistently better than all other methods at all positions, and PAVE is the second-best performer among other baseline methods.

nDCG@k

As explained earlier, nDCG captures the performance for graded relevance of venues. In terms of nDCG, DISCOVER is ahead of other state-of-the-art methods consistently in 9 sub-domains (CV, ML, IR, NLP, IP, WSN, AI, PDS, and SC) (Fig. 3.18 and Fig. 3.19) except SE, MM, and DM. DISCOVER shows consistent performances in terms of precision@k and nDCG@k in 9 sub-domains out of 12 sub-domains in CS.

For BIO, (Fig. 3.20), DISCOVER defeats other state-of-the-art methods consistently in 5 sub-domains such as CB, AN, BI, TX, and PL except for IM where it shows an average

Table 3.7: Overall precision (P@k) results (CS)

Methods	P@3	P@6	P@9	P@12	P@15
FB	0.541	0.581	0.583	0.585	0.590
CF	0.578	0.574	0.584	0.598	0.604
CN	0.615	0.625	0.629	0.631	0.632
CBF	0.634	0.618	0.612	0.605	0.609
RWR	0.638	0.620	0.615	0.613	0.615
CF+CBF	0.659	0.642	0.641	0.630	0.627
PVR	0.689^{+}	0.666	0.643	0.631	0.622
PAVE	0.666	0.671^{+}	0.648^{+}	0.635^{+}	0.637^{+}
DISCOVER	0.778*	0.733*	0.697*	0.685*	0.684*

'*' denote statistically significant results over the second best ('+')

Table 3.8: Overall precision (P@k) results (BIO)

Methods	P@3	P@6	P@9	P@12	P@15
FB	0.527	0.541	0.551	0.549	0.561
CF	0.540	0.549	0.556	0.557	0.563
CN	0.598	0.606	0.605	0.604	0.611
CBF	0.639	0.628	0.627	0.622	0.622
RWR	0.652	0.641	0.620	0.612	0.611
CF+CBF	0.653	0.654	0.641	0.634	0.626
PVR	0.670	0.661	0.637	0.629	0.632
PAVE	0.718^{+}	0.682^{+}	0.656^{+}	0.645^{+}	0.641^{+}
DISCOVER	0.794*	0.753*	0.727*	0.717*	0.706*

'*' denote statistically significant results over the second best ('+')

performance.

In terms of nDCG@k, both DISCOVER (Elsevier) and DISCOVER (Springer) do much better their counterparts, namely EJF and SJS in 10 sub-domains (MM, SC, ML, IP, DM, IR, WSN, SE, PDS, and AI) other than SC and CV in the CS domain (Fig. 3.21 and Fig. 3.22).

For BIO, DISCOVER (Elsevier) excels over EJF in all 6 sub-domains while DISCOVER (Springer) outranks SJS in 4 sub-domains (CB, BI, IM and AN) except TX and PL (Fig. 3.23).

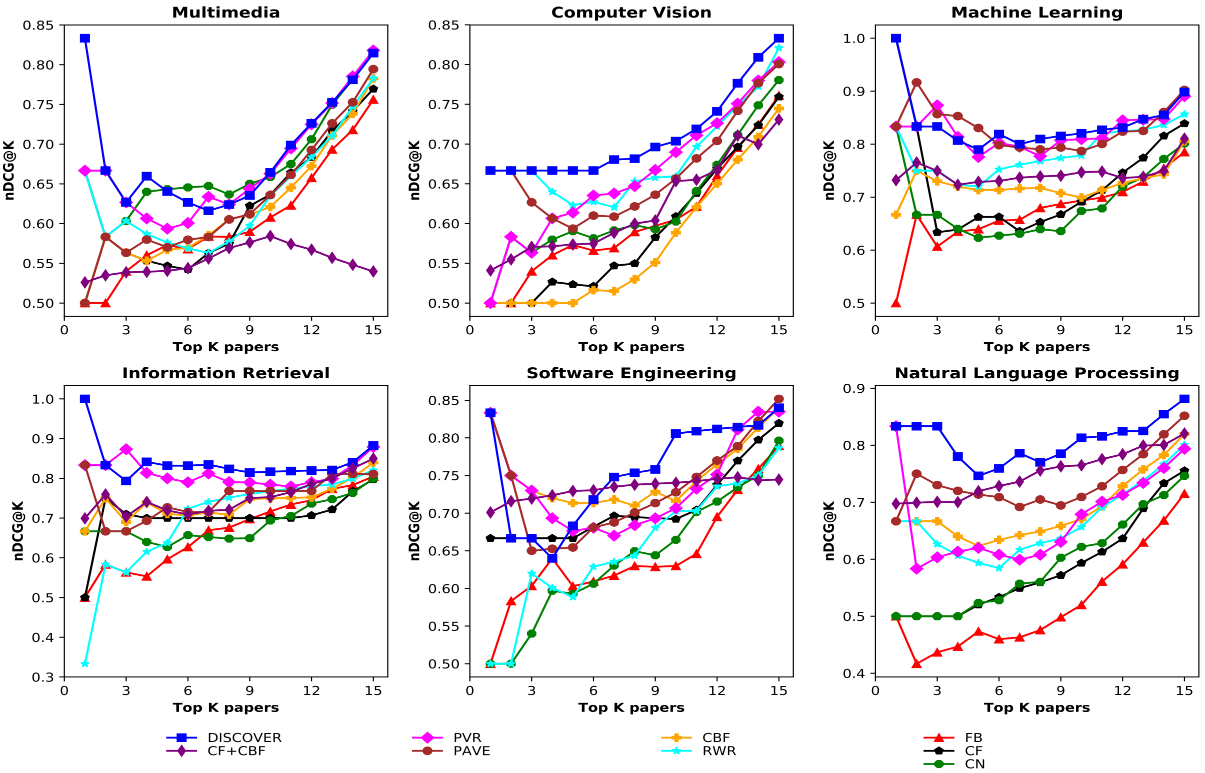


Figure 3.18: Sub-domain wise nDCG@k calculation (CS)

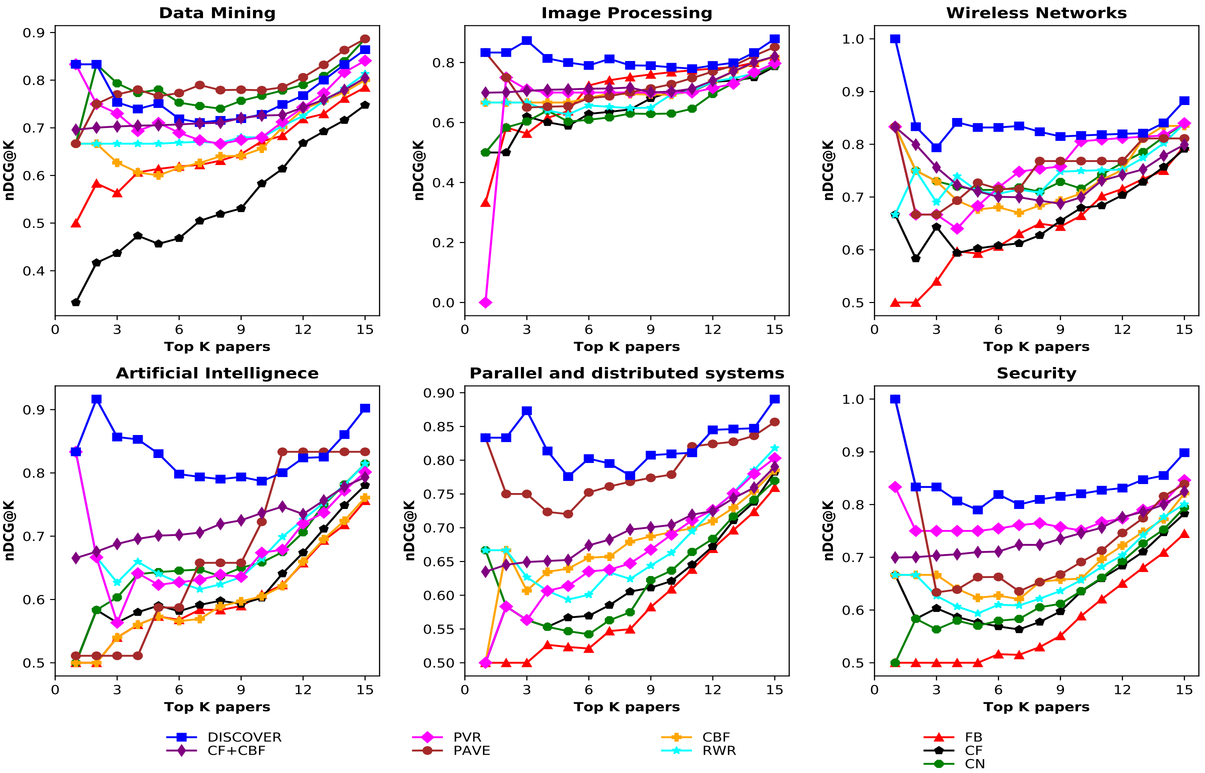


Figure 3.19: Sub-domain wise nDCG@k calculation (CS)

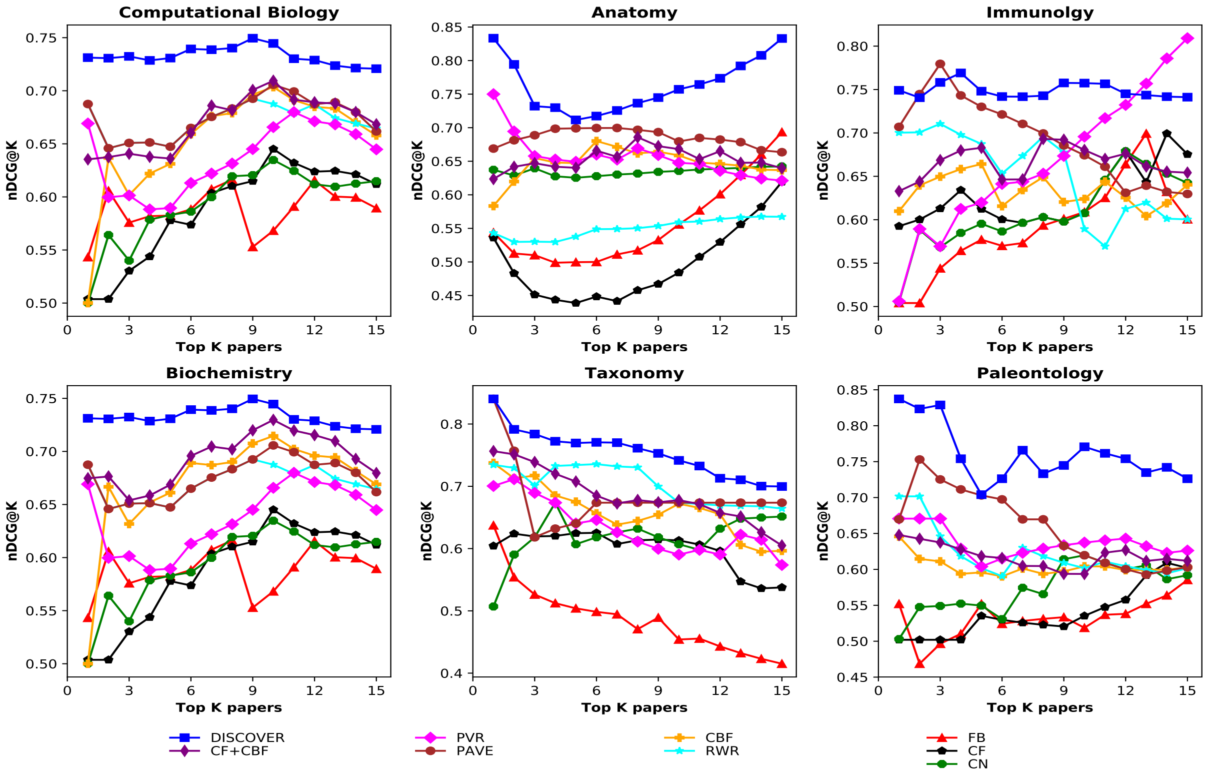


Figure 3.20: Sub-domain wise nDCG@k calculation (BIO)

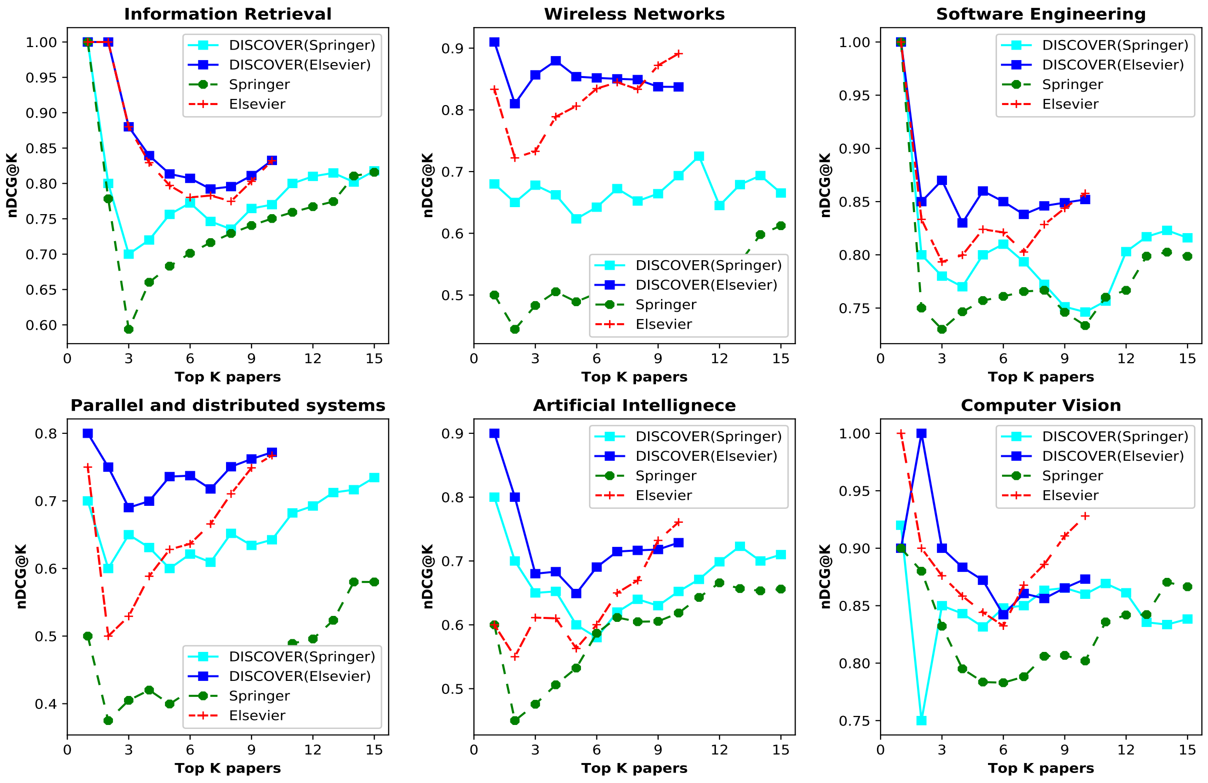


Figure 3.21: Sub-domain wise nDCG@k calculation (CS)

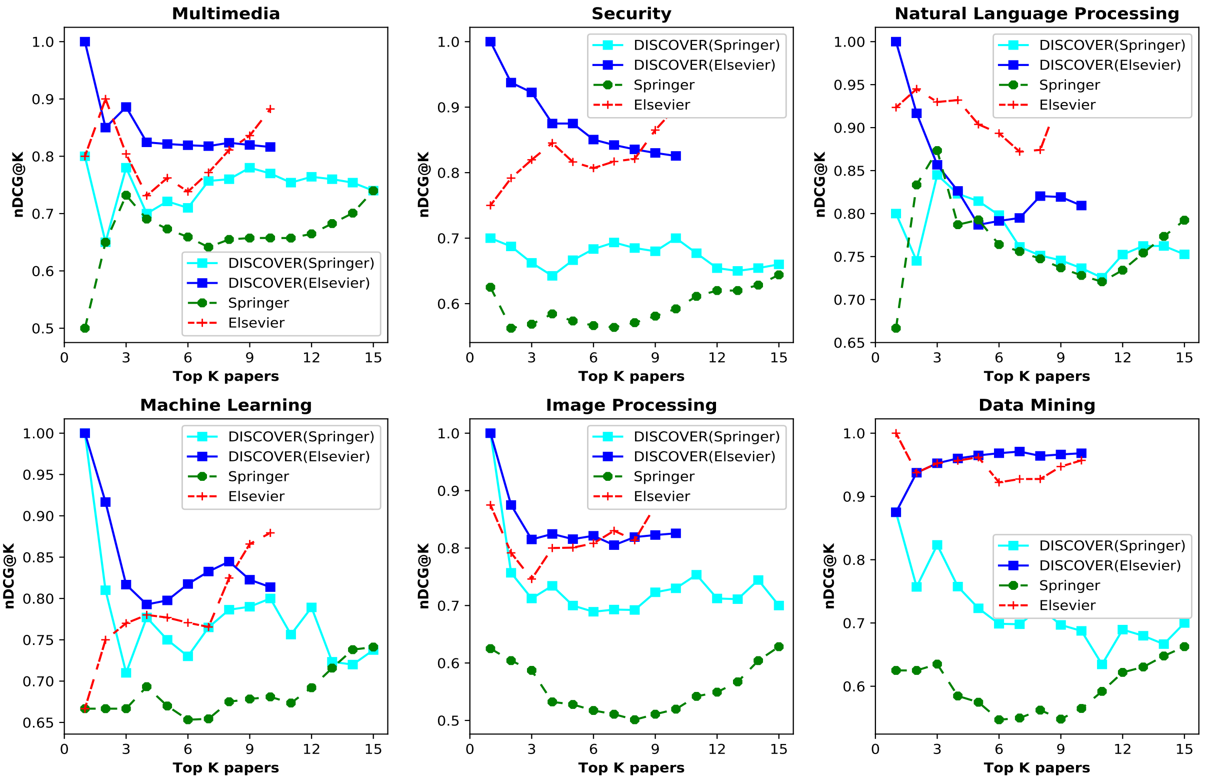


Figure 3.22: Sub-domain wise nDCG@k calculation (CS)

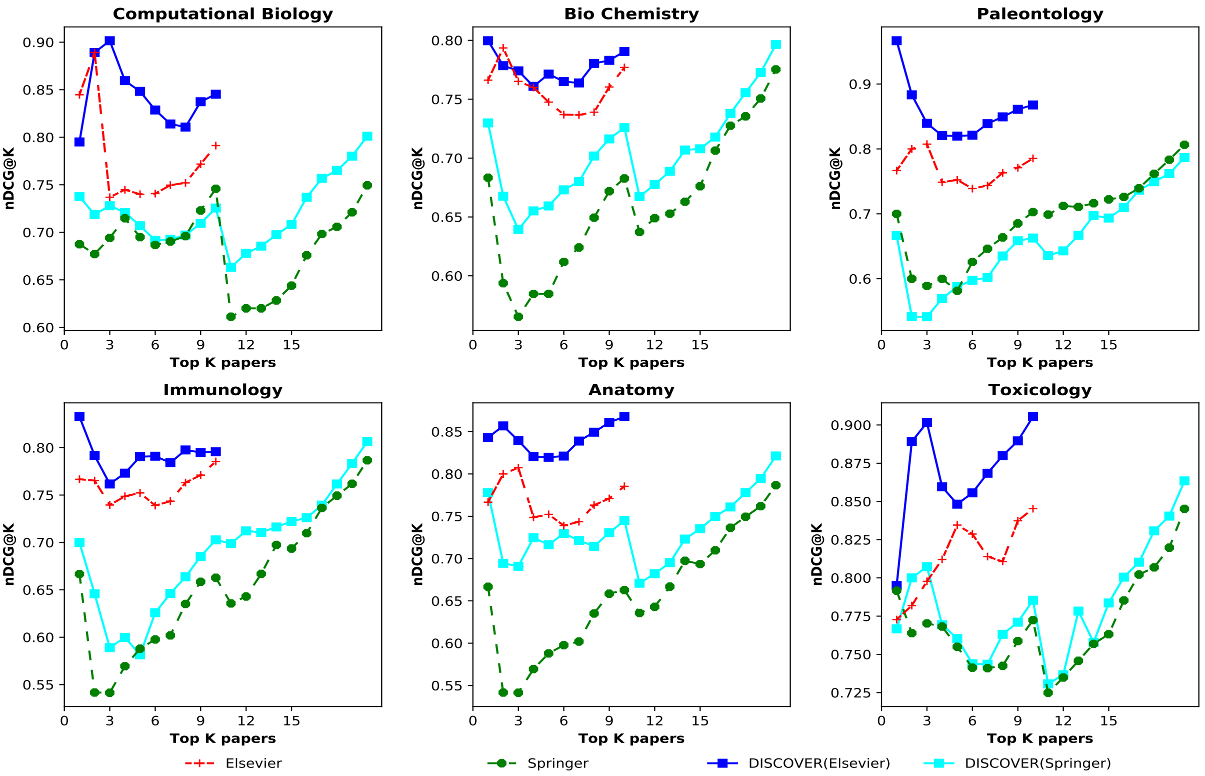


Figure 3.23: Sub-domain wise nDCG@k calculation (BIO)

Table 3.9: Overall nDCG@k results (CS)

Methods	nDCG@3	nDCG@6	nDCG@9	nDCG@12	nDCG@15
FB	0.541	0.586	0.622	0.687	0.772
CF	0.583	0.588	0.625	0.693	0.784
CN	0.616	0.624	0.647	0.712	0.802
CBF	0.643	0.640	0.664	0.717	0.804
RWR	0.644	0.646	0.674	0.735	0.812
PVR	0.687	0.686	0.702	0.753	0.828
CF+CBF	0.688$^+$	0.691$^+$	0.709	0.730	0.779
PAVE	0.672	0.688	0.714$^+$	0.771$^+$	0.840$^+$
DISCOVER	0.783*	0.765*	0.770*	0.802*	0.872*

'*' denote statistically significant results over the second best ('+')

Overall Results of nDCG@k

Average nDCG@k over 12 sub-domains of CS are depicted in Table 3.9. DISCOVER exceeds all other baseline methods with PAVE being the second-best except at position 3 and 6 (nDCG@3 and nDCG@6). CF+CBF performs better than PAVE only at positions 3 and 6. Afterward its shows a lower nDCG than both PVR and PAVE methods. Among the rest, RWR performs better than CBF, CN, CF, and FB with FB being the worst performer.

Table 3.10: Overall nDCG@k results (BIO)

Methods	nDCG@3	nDCG@6	nDCG@9	nDCG@12	nDCG@15
FB	0.538	0.544	0.543	0.579	0.578
CF	0.541	0.558	0.571	0.601	0.609
CN	0.575	0.589	0.617	0.628	0.626
CBF	0.656	0.658	0.666	0.657	0.642
RWR	0.648	0.642	0.654	0.637	0.627
CF+CBF	0.664	0.661	0.675	0.669$^+$	0.643
PVR	0.631	0.631	0.642	0.657	0.653$^+$
PAVE	0.685$^+$	0.686$^+$	0.678$^+$	0.660	0.648
DISCOVER	0.760*	0.742*	0.752*	0.743*	0.740*

'*' denote statistically significant results over the second best ('+')

In BIO, there is a clean sweep for DISCOVER in terms of nDCG irrespective of positions (Table 3.10). PAVE exhibits the second-highest performance.

Table 3.11: Diversity (D) and Stabilty (MAS) of DISCOVER and other approaches

Methods	D (CS)	D (BIO)	MAS (CS)	MAS (BIO)
FB	0.227	0.241	9.961	9.864
CF	0.387	0.355	8.936	8.873
CN	0.281	0.274	9.784	9.862
CBF	0.219	0.206	5.887	6.045
RWR	0.312	0.322	8.992	9.137
CF+CBF	0.394	0.369	5.639^{+}	5.582^{+}
PVR	0.403^{+}	0.397^{+}	8.236	8.179
PAVE	0.327	0.319	8.863	8.761
DISCOVER	0.519*	0.503*	4.758*	4.695*

'*' denote statistically significant results over the second best ('+')

3.5.3 Evaluation of Diversity

Diversity is defined in terms of content dissimilarity. We group all papers published at a particular venue and extract their corresponding keywords. We apply the similarity score in Eqn. 2.29 in the definition of diversity (Table 3.11). DISCOVER is seen to show the best diversity, and its performance gap with the second-best (PVR) is statistically significant (at 5% level) for both CS and BIO as shown in Table 3.11.

3.5.4 Evaluation of Stability

We have also provided a comprehensive investigation of the stability of the proposed DISCOVER, as defined in Eqn. 2.30. DISCOVER shows the minimum MAS than all other standard approaches (Table 3.11). It shows a MAS of 4.758 for CS, meaning that on an average, every predicted venue will shift by a position of 4.758 after adding new data into the training data of the system. Similarly, it shows a MAS of and 4.695 for BIO. We have considered the average MAS-score as a threshold to decide whether a particular method provides stability or not.

3.5.5 Ablation Study and Analysis

We also conduct an ablation study to showcase the contribution of different components (major modules) on the performance of DISCOVER from different perspectives. The precision, MRR, accuracy, nDCG, diversity, stability and H5-Index obtained by DISCOVER

Table 3.12: Ablation study on DISCOVER

Method	Acc@15	MRR	P@15	nDCG@15	Diversity	MAS	H5-Index
D-KBS	0.476$^+$	0.084	0.378$^+$	0.476	0.481	**7.924**	89
D-SNA	0.589	0.128	0.469	0.594	**0.296**	4.986	69$^+$
D-Title	0.631	0.152	0.601	0.798	0.508	5.321	93
D-Keyword	0.512	0.103	0.435	0.567	0.506	5.872	91
D-CA	0.497	0.082$^+$	0.393	0.473$^+$	0.318$^+$	4.935	79
D-MPA	0.612	0.149	0.511	0.695	0.417	6.851	**67**
D-Abstract	**0.409**	**0.069**	**0.361**	**0.451**	0.506	7.693$^+$	86
DISCOVER (D)	0.713	0.163	0.679	0.864	0.508	4.631	98

The top-most contributing components and the second-best are marked by 'bold-face' and '+' respectively

and when one component is removed therefrom at a time are shown in Table 3.12. In the Method column, KBS, SNA, CA, and MPA denote keyword-based search strategy, social network analysis, citation analysis, and main path analysis methods, respectively. The rows represent performance of the entire DISCOVER system and when a module is removed from DISCOVER in turn. Acc, MRR, and P denote Accuracy, Mean Reciprocal Rank, and Precision.

More significantly different components of the model have different effects on various assessment steps. As can be seen from the table, the use of keyword-based strategy and the similarity of abstract-based model is very important: if it is withdrawn, both accuracy and precision drop dramatically by almost 30 percent. Citation analysis is also a necessary component that contributes the most to achieving the model's MRR and nDCG as evident from the largest drop in nDCG. Similarly, the largest drop happens for diversity when the social network analysis module is removed. We find that eliminating a search strategy based on a single component keyword from the model results in drops of up to 3.3 MAS points across both CS and BIO domains. When the main path analysis module is removed, we see a similar decrease in the overall standard of venues in terms of the H5-Index. These findings illustrate the importance of, and, thus justify combining multiple modules from different contexts in DISCOVER.

The objective of DISCOVER is to reduce total computation cost in recommendation besides addressing the issues of sparsity, diversity, and stability. Hybrid binary tree architecture with a keyword-based search strategy is adopted in the initial stage. In a variety of ways, we use link and content similarity-based methods to boost the model's relevance.

Social network analysis using different centrality measures and content characteristics of a paper such as title and keywords are used in the proposed system to capture important terms and relevance of other papers. Main path analysis, which tracks the most significant paths in a citation network, is used to provide conceptually similar papers to a given seed paper. Finally, abstract matching is performed only on a small set of papers carefully filtered through a pipeline of steps. Total computational overhead does not, therefore, substantially increase with increase in the number of papers. Each module has its own significance and they complement each other in a cascaded manner to generate the final recommendation.

3.5.6 Study of the Proposed Approach

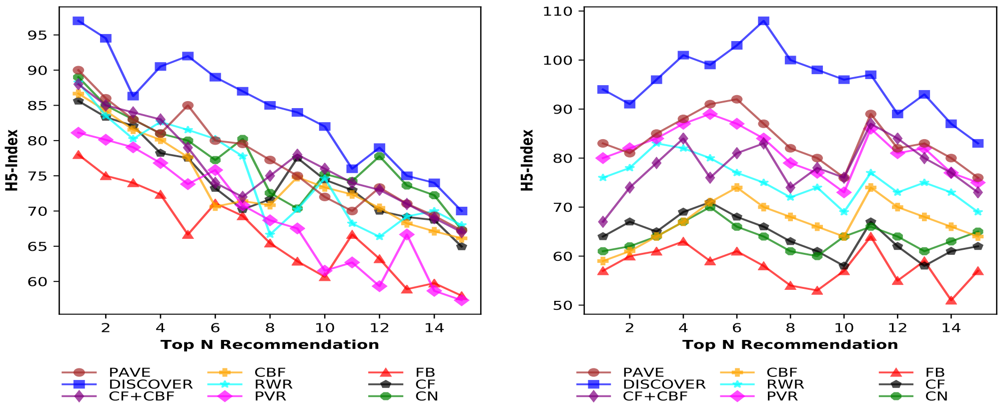
Here we revisit the sub-queries (SQs) pertaining to the broad RQs that we started with along with our observations.

SQ1: How Effective is DISCOVER in Comparison to Other State-Of-The-Art Methods?

We see the performance of DISCOVER vis-a-vis the other methods for each of the sub-domains of CS and BIO dataset in detail (Fig. 3.12, Fig. 3.13, Fig. 3.14, Fig. 3.15, Fig. 3.16,Fig. 3.17, Fig. 3.18, Fig. 3.19, Fig. 3.20, Fig. 3.21, Fig. 3.22, Fig. 3.23). Also we measure the overall performance of DISCOVER over all sub-domains taken together. The overall results of precision@k, nDCG@k, accuracy and MRR as shown in Tables 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10 are statistically significant (paired-samples t-test at α=0.05) over other approaches in both the domains of CS and BIO.

SQ2: How is the Quality of Venues Recommended by DISCOVER?

The venues recommended by DISCOVER are of high quality as compared to other state of the art methods. including EJF and SJS recommendations (Fig. 3.24a, Fig. 3.24b, Fig. 3.25a and Fig. 3.25b). The average H5-index of DISCOVER shows the highest average value of 97 while recommending first venues, 86 while recommending the 3rd venue, then slightly downgrades and ends with a H5-index of 70 at position 15 in domain CS as depicted in Fig. 3.24a. In BIO, DISCOVER shows the highest average H5-index of 108

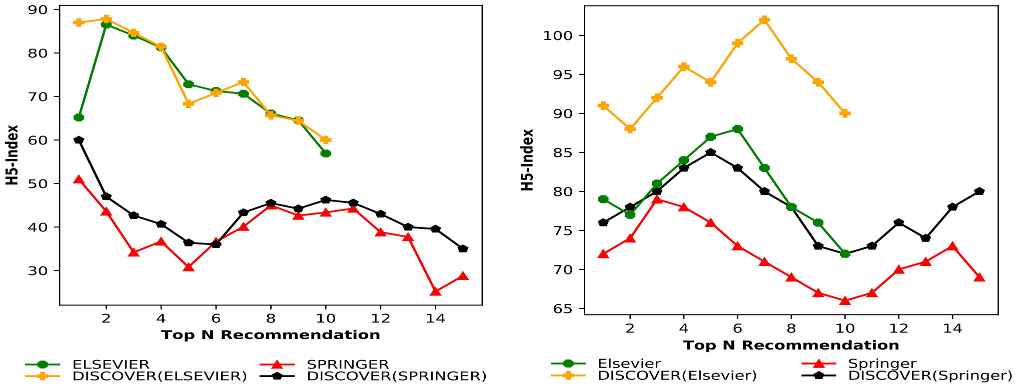


(a)

(b)

Figure 3.24: (a) Average venue quality (CS) (b) Average venue quality (BIO)

while recommending the 7th venue as shown in Fig. 3.24b. DISCOVER (Elsevier) recommending similar venues with EJF except for few positions (1st and 7th) recommendation in the domain of CS where DISCOVER is better (Fig. 3.25a). The average H5-index of DISCOVER (Elsevier) is 75 whereas the average H5-index of EJF is 71. Similarly DISCOVER (Elsevier) performs better than EJF with an average H5-index of 94 in the domain of BIO (Fig. 3.25b).



(a)

(b)

Figure 3.25: (a) Average venue quality of EJF and SJS(CS) (b) Average venue quality of EJF and SJS (BIO)

Table 3.13: MRR results of proposed DISCOVER and other approaches

Approach	MRR					
	$2<=v_c<8$	$8<=v_c<15$	$15<=v_c$	$2<=p_c<8$	$8<=p_c<15$	$15<=p_c$
FB	0.017	0.025	0.029	0.019	0.023	0.027
CF	0.024	0.026	0.030	0.022	0.027	0.029
CN	0.026	0.029	0.035	0.023	0.032	0.030
CBF	0.032	0.039	0.038	0.027	0.037	0.038
CF+CBF	0.038	0.049	0.056	0.044	0.043	0.046
RWR	0.039	0.046	0.048	0.028	0.038	0.041
PVR	0.042	0.051	0.056	0.035	0.042	0.044
PAVE	0.096^{+}	0.108^{+}	0.115^{+}	0.096^{+}	0.104^{+}	0.109^{+}
DISCOVER	0.147*	0.176*	0.180*	0.164*	0.169*	0.171*

'*' denote statistically significant results over the second best ('+')

SQ3: How does DISCOVER Handle Cold-start Issues for New Researchers and New Venues and Other Issues?

The overall comparison on various issues, including cold start issues, are listed in Table 3.15.

(i) **Cold-start Issues**: We take the average MRR of both CS and BIO in all 6 categories mentioned in Sec. 3.4.4. The analysis in Table 3.13 shows that, even if the seed paper related to a new venue and new researcher, DISCOVER could predict the original venue at early ranks. It does not require past publication records or co-authorship networks for the recommendations. It considers only the current area of interest along with the title, keywords, and abstract as inputs to recommend the same.

Table 3.14: Step-wise papers filtration of both CS and BIO

Steps	No. of papers (CS)	No. of papers (BIO)
Original Dataset	15,641,658	14,785,486
Data preprocessing (Training)	10,424,960	9,961,893
Keyword-based search	5k-13k	4k-11.5k
Centrality measure calculation	2k-5k	1.5k-4k
Co-citation score computation	800-2k	500-1.5k
Main path analysis	45-85	55-95
Abstract Matching	60-100	70-110

(ii) **Data Sparsity**: To specifically address data sparsity issue, social network analysis through various centrality measures and content features of a paper like abstract,

Table 3.15: Issues involved in DISCOVER and other compared approaches

Methods	Cold-start	Sparsity	Diversity	Stability
FB	yes (new researcher)	no	yes	yes
CF	yes (researcher and venue)	yes	no	yes
CN	yes (new venue)	no	yes	yes
CBF	yes(new venue)	no	yes	no
RWR	yes (new researcher)	no	yes	yes
CF+CBF	yes (researcher and venue)	yes	no	no
PVR	yes (researcher and venue)	yes	no	yes
PAVE	yes(new researcher)	no	yes	yes
EJF	yes(new venue)	no	yes	no
SJS	yes(new venue)	no	yes	no
DISCOVER	no	no	no	no

title, and keywords were exploited to capture the strength of both significance and relevance, respectively. It has been observed that the average number of papers found after keyword matching are in the range of 5k-13k and 4k-11.5k for CS and BIO, respectively. Table 3.14 displays the stepwise filtration of papers of both CS and BIO. After the initial filtering, we are left with meaningful papers for further computation, which are close to the area of interest. Hence there is no data sparsity issue in our proposed approach, as mentioned in Table 3.15.

(iii) **Computational Costs**: In DISCOVER reduction of computational costs has been prioritized. It applies the main path analysis to extract only relevant and conceptually related papers. As shown in Table 3.14, there is more than 90% reduction after the initial step of the keyword-based search, and there is a substantial reduction in further steps as well. Table 3.14 shows the average number of papers involved in each step of the proposed approach. We believe that the proposed system will show a satisfactory performance with a larger dataset. Even a substantial increase in dataset size will not impact the overall computation time by much and therefore does not suffer from scalability issue.

(iv) **Diversity**: Several steps are taken to ensure diversity in the result set. We use both link and content similarity-based techniques at a number of places. Also, the main path analysis, that traces the most significant paths in a citation network captures conceptually related papers to a given seed paper. Integration of these approaches can provide recommendations from diverse publishers, as evidenced by Table 3.11. DISCOVER shows the highest value of D (diversity) as compared to all other approaches.

(v) **Stability**: We build a content-aware recommender system based on the title, keywords, and abstract similarities. During the initial stages, centrality measures are calculated, and thereafter, the textual content similarity is computed to find the related papers. Ranking of venues is done at a very later stage from a collection of related papers filtered out in a long pipeline. The addition of new papers, therefore, do not affect the order of recommendations. In all these batteries of techniques together provide stability to the recommendations. DISCOVER shows the minimum MAS than all other standard approaches (Table 3.11).

3.5.7 Some Insights

Overall good scores discussed in Sec. 3.5.2 and Sec. 3.5.1 showcase the efficacy of the proposed DISCOVER for venue recommendation. However, there are few limitations as follows.

(i) The proposed system may not recommend relevant venues with less than 3 to 5 domain-specific keywords. As a result, DISCOVER displays average performance in terms of precision@k in few sub-domains like SE, MM, and DM as depicted in Fig. 3.12 and Fig. 3.13.

(ii) If there are an insufficient number of related papers, the proposed system may fail to capture the relevant papers resulting in possibly irrelevant venue recommendations. Due to these constraints, DISCOVER exhibits the worst performance of precision against EJF in sub-domains ML, DM and against SJS in sub-domains IR and SE as depicted in Fig 3.12 and Fig. 3.13.

(iii) The proposed approach displays the worst nDCG against EJF as depicted in Fig. 3.22. The minimum number of related papers of a specific sub-domain required for good recommendation is found to be in the range of 1k-2k.

(iv) The proposed system hence exhibits the worst nDCG against both EJF and SJS in sub-domain CV as depicted in Fig. 3.22. The system could recommend relevant venues if the citation network is strongly connected.

3.6 Conclusions

Academic venue recommendation is an emerging area of research in recommendation systems. The set of proposed techniques are few in numbers, and they suffer from several problems. One of the major issues is that of cold-start having two sub-parts: that for new venues and new researchers. Also, there exist other issues of sparsity, diversity, and stability that are hitherto not adequately addressed by existing state-of-the-art methods. This paper proposes a diversified yet integrated social network analysis and contextual similarity-based scholarly venue recommender (DISCOVER) system that reasonably addresses all the above-mentioned issues. It is developed taking into account recent advances in social network analysis incorporating centrality measure calculation, citation and co-citation analysis, topic modeling based contextual similarity, and main path analysis of a bibliographic citation network.

To assist in identifying relevant research outlets, contextual similarity through a hybrid approach of both topic modeling and matrix factorization techniques are adopted. We conducted an extensive set of experiments on a real-world dataset: MAG, and demonstrated that DISCOVER consistently outperforms the state-of-the-art methods and other freely available online services such as EJF and SJS. On two different domains of field (CS and BIO), DISCOVER shows significantly better scores of precision@k, nDCG@k, accuracy, MRR, $F - measure_{macro}$, diversity, and stability than other state-of-the-art methods. DISCOVER also suggests high-quality venues as compared to state-of-the-art methods and other freely available online services such as EJF and SJS in terms of H5-index. Nonetheless, there is scope for future study in this direction. We plan to experiment with other datasets and to extend it to multiple disciplines with the goal of improving accuracy, novelty, coverage, and serendipity. We would also like to investigate the same with the help of heterogeneous bibliographic information network with meta path features.