**4. MODEL SPECIFICATION AND ANALYSIS**

SNAVER is designed for shortlisting academic venues to make personalized recommendation for researchers. The model is inspired by the fact that, a good researcher generally desires to: get in contact with academic venues which acknowledges high quality and fruitful papers, participate in academic conferences that are closely related to their research, and contribute to those venues where it is possible for them to publish (.) their research papers and achievements.

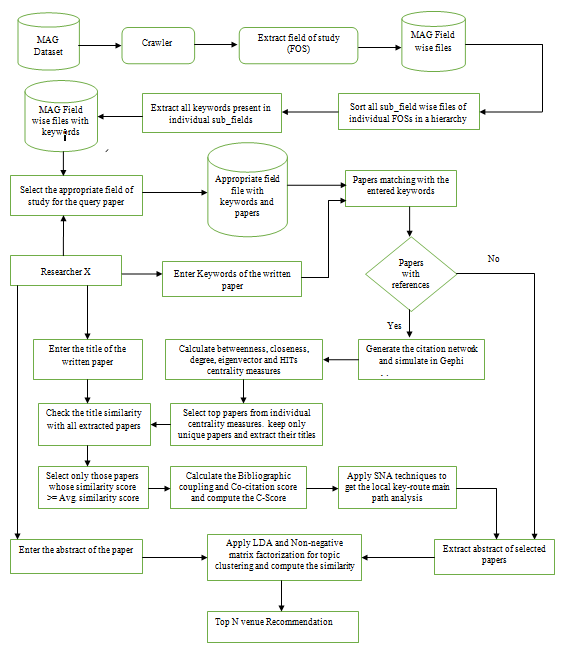
**4.1 Design of SNAVER**

Additionally, SNAVER evolves from the basic social network analysis model which has been proved to be suitable for calculating the similarity of nodes in networks and to measure the importance of individual nodes. Factors like betweenness, closeness, degree, eigenvector, hub and authority scores aim at biasing the citation network, to identify the most influential nodes. Traversal counts like search path count (SPC), search path link count (SPLC), search path node pair (SPNP), and other variations [**A new approach for main path analysis: Decay in knowledge diffusion**] measure the significance of each link. SNAVER considers the relationship among the papers in a broad area and evaluates the significance of each paper through certain centrality measures. Centrality identifies the most important papers within the network. The process for the proposed SNAVER approach comprises three steps: (1) generation of multilevel citation networks, (2) selection of candidate papers, (3) selection of most significant papers with the help of key route selection, and (4) determination of ranking of each selected significant paper with the help of abstract similarity checks for final recommendation. The detail process of SNAVER is described below. The architecture of our SNAVER model is illustrated in figure 2.

**4.2 Model Specification**

Any researcher containing title, keywords and abstract for her paper can use our system to get suggestions about suitable academic venues for the same. The detailed process of our proposed Social Network Analysis based Venue Recommender system (SNAVER) is described below:

* *Step 1: Data Collection* Initially we constructed the dataset by crawling data from Microsoft Cognitive Services Academic Knowledge API. We further processed the dataset by storing papers and their keywords as per the field of study they belong to. Each Field belongs to some level (level1, level2 or level3). Higher the level more general or wide the field is. Also, we have confidence score between fields, which is a score signifying how related the 2 fields are.
* *Step 2: Data Pre-Processing.*Created a hybrid binary tree, i.e., fields in each level are divided in 2 parts based on the average of the confidence score between the parent and all its child fields (fields which have a confidence score between them and lie in the level below the parent field level). Recursively all the keywords of the children are fetched and stored along with the parent node's keywords. Each field acts as a node containing 5 things: (1) left subtree nodes (2) left subtree keywords set (all keywords in all the left subtree nodes recursively fetched till the end) (3) right subtree nodes (4) right subtree keywords set (same as above for right subtree) (5) its own keywords. (Figure 3)
* *Step 3: Extraction of papers.* The input data is a set of publications with title, abstract and keywords grouped with respect to their field of study. Initially the researcher selects the appropriate field of study from the list given in the system. The system extracts all papers which belong to this domain. Next, the system will ask a few related keywords of the paper, for which the researcher wants to know the venue. Based on these keywords, the matching group or groups of publications are shortlisted, and all the unique papers of these groups are extracted.
* *Step 4: Check for the availability of references.* Now, to build the citation network between the papers, the system will check whether the extracted paper’s references are there in the dataset or not. It will divide them in two sets. It will keep those papers whose references are available in the dataset in one set (set-I) and the other set (set-II) will retain the other papers whose references are not available.
* *Step 5: Forming citation network and computation of various centrality measures.* The system will proceed with set-I. It will make a citation network of all papers from that set. The 5 centrality measures: betweenness, closeness, degree, eigenvector and hits score of/for all individual papers are calculated. Individual average of each measure is used as a threshold to further narrow down the number of candidate papers.
* *Step 6: Calculation of title similarity and computing C-Score.* SNAVER now asks for the title of the seed paper. This title is compared with all the titles of filtered papers from the above step. All papers whose title similarity score is greater than the average similarity score is retained. And only those filtered papers are taken into consideration. The bibliographic coupling (BC) and co-citation (CC) score can be computed to get the more similar papers and to get a C-score to filter those papers which are not that much related to the paper of interest (POI). We need to retain only those papers which are satisfies a minimum of average C-score and can be taken into further computation.
* *Step 7: Computation of main path analysis through identification of key route.* The filtered papers are then again used to make the citation network. This time the significance of each link is considered by giving weights according to the search path node pair algorithm which is the number of times the link is traversed if one runs through all possible paths from all the ancestors of the tail node (including itself) to all the descendants of the head node (including itself). [Hummon, Norman P.; Doreian, Patrick. [*"Connectivity in a citation network: The development of DNA theory"*](https://doi.org/10.1016/0378-8733(89)90017-8). Social Networks. **11** (1): 39–63. [*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier):[*10.1016/0378-8733(89)90017-8*](https://doi.org/10.1016%2F0378-8733%2889%2990017-8).] Based on these weights Key-route search is performed to select the most influential nodes/papers from the citation network.
* *Step 8: Selection of candidate papers to* check abstract similarity. After getting the key route papers in the previous step, the recommender system will add a few more papers whose references were not available in the dataset. Title similarity is performed for set-II from step-4 and the top similar papers are chosen. We are doing this due to the high title similarity of the seed paper with them. We have two assumptions regarding the inclusion of these papers for abstract similarity check. 1) There may be a few papers present whose references are not there in the dataset but may be involved with many reputed venues. 2) The seed paper’s title is matching with them so there is a probability that the seed paper may get accepted with the same level of venues.
* Step 9: C*omputing abstract similarities by LDA and non-negative matrix factorization (NMF) methods.* After selecting all candidate papers, we perform the abstract similarities based on two methods. Firstly, LDA technique has been used to calculate the similarity of individual papers’ abstract with the seed paper’s abstract. Secondly, this similarity is again calculated using Non-negative matrix factorization.
* Step 10: *Computing the Rank and suggesting the journal*. Each paper is given a rank based on their score of similarity with the seed paper’s abstract using both LDA and NMF. Finally, both these ranks are combined to get the final rank of the papers. The ranked papers are used to fetch the journals in the same order and suggest the top k (specified by the user) unique journals.

**

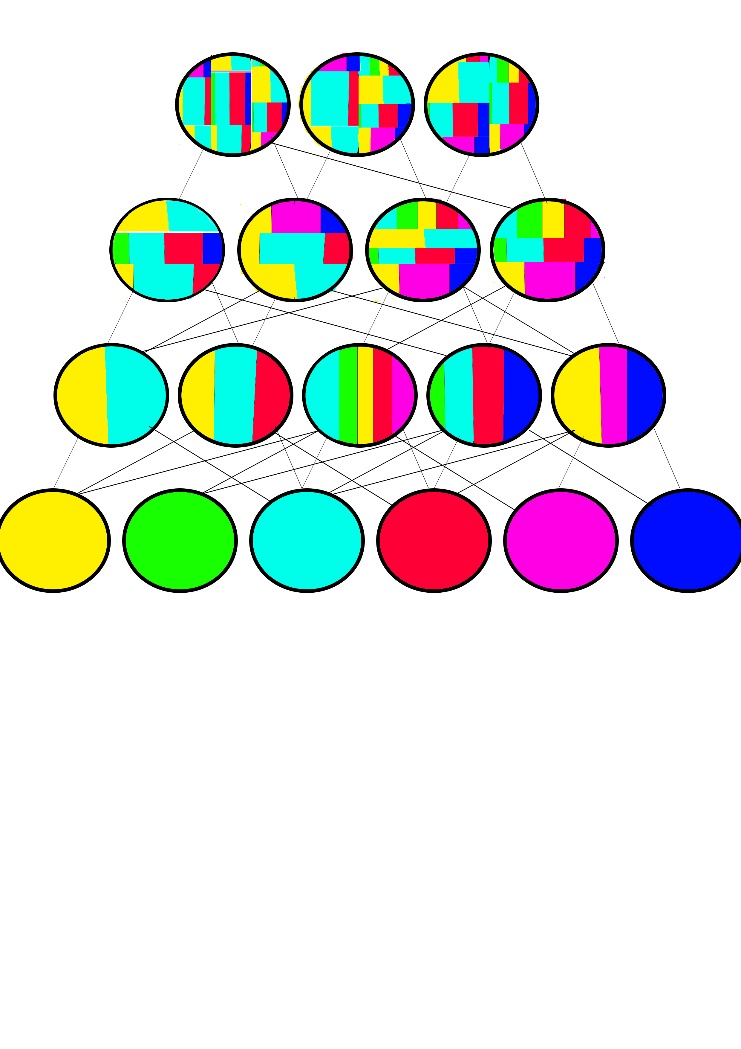
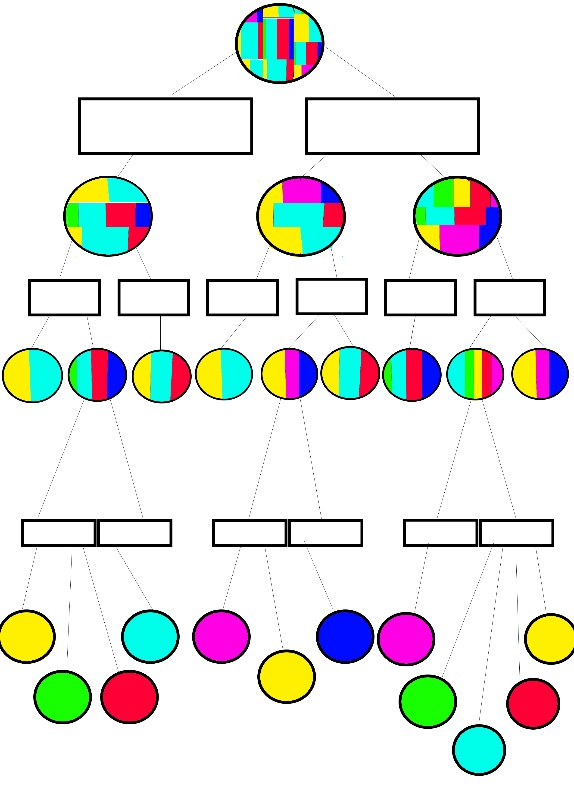
**Figure 2: The structure of SNAVER**

* *Step 10: Final venues recommendation by merging the results.* The system stores the order of the recommended venues using both LDA and non-negative matrix factorization methods. Later, the ranking of both these methods are merged to make an efficient recommendation. It takes the sum of the rankings of these two methods and divide it by 2 to produce a final ranking of venues. The same ranking would be recommended by the proposed system.
* *Step 11:* The target researcher can give a few inputs such as venues with specific publishers, only journal, only top tier conferences, minimum impact factor of the journals, number of suggested venues etc to refine the results to make it personalised venue recommendation.

**4.3 Model Analysis**

**4.3.1 *Selecting the field of study and making the hybrid graph***

Information about which field or fields of study does a publication belong to is very valuable for many tasks. The MAG dataset contains papers, their related keywords and its field of study. For each field of study, we have stored all the papers and keywords related to that field in one file. Each Field belongs to some level (level0, level1, level2 or level3). The list given to the researcher are all the level 0 fields; computer science, physics, chemistry, biology etc. (19 such fields exists). These are very diverse fields and the user selects one of these fields. Figure 3 illustrates how the actual fields are present and related to each other.

** **

**Figure 3 Figure 4**

**Figure 3 represents the actual hierarchy of the dataset. Colors can be considered as papers. Papers belong to one of the level 3 nodes and these nodes can be a part of multiple level 2 nodes which can be a part of multiple level 1 nodes. Although the number of level 1 nodes who are a part of multiple level 0 nodes are comparatively less, it is also possible. (In case of inter disciplinary fields) Hence, locating the field of study to get all the relevant papers using the keywords is a very tedious and difficult task.**

**The hybrid binary tree (figure 4) divides each level in 2 halves, one greater than the average confidence score and the other equal to and less than the average confidence score of all the children nodes with the parent node. The keywords of all the papers are fetched from the bottom to top most level and stored after grouping in the higher levels. There is duplication of data, but the amount of computation time is saved largely as at every step just like the binary tree search algorithm, the unmatched half side of the tree is pruned.**

The main purpose is to select all the papers from the fields which are relevant to the user, based on the keywords entered. This is done by checking similarity of all field’s keywords with that of the entered keywords. choose the fields which are most similar and extract all its papers. To optimize this search a hybrid binary tree is created as discussed in the step 3 of data pre-processing. Figure 4 shows an example of how the tree looks like.

*4.3.1 Search using keywords*

The fields related to each other have a confidence score given to them. Score of 1 implying the 2 fields are very similar (part of or dependent on each other) and lower scores implies lesser similarity. If the 2 fields are not similar at all, their confidence score will be 0. To select all the papers from the field which is relevant to the user based on the keywords entered. A queue Z is created which at start contains only the node corresponding to the field of study chosen by the user. Each node is popped from this queue Z. The given set of keywords is matched with the left, right and parent sets of keywords of the popped node separately. Upon match, the number of matches is checked. Case 1: If the number of matches is greater on any one side, the other side is pruned. and all the nodes of the greater side are added to the queue Z. Case 2: If the number of matches on both the sides are equal, nodes from both the sides are added to the queue Z. 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 no more nodes are left in the queue or we have reached level 3. The scores (i.e., number of matches) for each node is compared and papers from all the nodes with maximum score are extracted.

Matching of the keyword is done in the following steps:

1. The given keywords are tokenized, and checked for stop words to be removed.
2. We use the Lancaster stemmer to get the root words.

These stemmed words are checked for equality for matching.

*4.3.2 Centrality measure calculation and its significance*

Different centrality measures determine the significance of individual papers with respect to their relationships with other papers [32]. We calculate five centrality measures (degree centrality, closeness centrality, betweenness centrality, eigenvector centrality and HITS score) to filter the non-significant papers.

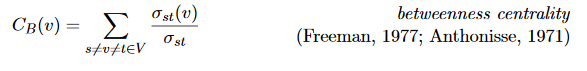
[Heymann S. (2014) Gephi. In: Alhajj R., Rokne J. (eds) Encyclopedia of Social Network Analysis and Mining. Springer, New York, NY]

The following are the centrality measures:

1. *Betweenness centrality (CB)*

It measures how often a node appears on shortest paths between nodes in the network (Brandes,2001: [ <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=B529D5F84AD2011762A5BE4BF29FED9E?doi=10.1.1.11.2024&rep=rep1&type=pdf> ] )*.* Vertices with high betweenness act as potential deal makers. Their position is very crucial as most of the other nodes are connected through them. In other words, this measure is the extent to which a paper is positioned on the shortest path between other pairs of papers.

In a citation network, papers with closely related citations form a community. Papers which link communities have the ability to control information flow among communities, and thus are important. Typically, research papers that have an impact on papers belonging to multiple fields or those that merge existing concepts tend to have a high score.



Let (sigma)st denote the number of shortest paths from s to t and (sigma)st(v) denote the number of shortest paths from s to t such that some v lies on it.

1. *Degree Centrality (CD):*

Degree centrality is the most intuitive notion of centrality [33]. The degree of a node is the number of edges that are adjacent to that node. More the number of neighbour a given node has, the greater its influence is. We consider both the number of papers “citing,” called the in-degree centrality and the number of papers “cited,” called the out-degree centrality:

CD (P) = indeg(P) + outdeg(P)

where indeg(P) is the number of papers referring to paper P and outdeg(P) is the number of papers P is referring to. A high value of the in-degree centrality implies popularity and high value of out-degree can be used to identify a well referenced paper. We remove the papers with low sum of in-degree and out-degree as those papers are not being cited nor do have any references and hence are assumed to be neither the best work nor published in a very influential journal. The calculation of the degree centrality is limited by the number of nodes that are directly connected to the paper, and indirectly connected nodes are not included for the measurement.

1. Closeness Centrality:

Closeness centrality is based on the distance from a node to all other nodes in the network, and is defined as the inverse total distance. It implies that a paper is more central if it interacts with more number of nodes, and it is considered relatively important [34].

It is the average distance from a given node to all other nodes in the network (Brandes 2001).



Let dG (v, t) to denote the distance between vertices v and t, i.e. the minimum length of any path connecting v and t in G.

1. Eigenvector Centrality:

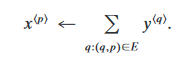
Node importance in a network based on a node’s connections. A node is central to the extent that the node is connected to others who are central. It depends on the number of neighbour nodes that are directly connected to a paper and the quality of the neighbour nodes [36].

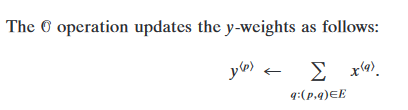


where AP,J is the adjacency matrix in which its element is one if J is linked to P, and zero otherwise. xJ is the score of the eigenvector centrality of J, and λ is the eigenvalue of P. Eigenvector centrality measures not only how many papers are connected to a paper, but also how many important papers are connected to a paper.

1. HITS:

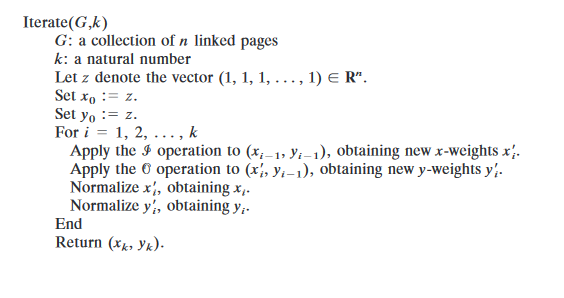
Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm. The HITS metric determines 2 values for a page: its *authority,* estimate of the value of the content/quality and its *hub value,* that is the estimate for the links to other papers. In other words, a good hub represents a paper that points to many other papers, and a good authority represents a paper that is linked by many different hubs.

Given weights {x^p}, {y^p}, the  operation updates the x-weights as follows: 



the set of weights { x ^ p } as a vector x with a coordinate for each page in G s ; analogously, we represent the set of weights { y ^ p } as a vector y.

[http://delivery.acm.org/10.1145/330000/324140/p604-kleinberg.pdf?ip=1.186.14.57&id=324140&acc=ACTIVE%20SERVICE&key=045416EF4DDA69D9%2E517DE04875AE9835%2EE9E06D43DBF1A2CE%2E4D4702B0C3E38B35&CFID=1026961323&CFTOKEN=22443355&\_\_acm\_\_=1515910385\_83f6e3e085919dcd9a750071ca3efc6a]



We select papers based on each of the centrality measures mentioned above. Average score of each measure is calculated and a list containing papers greater than this average is stored. Thus, obtained 5 lists are then merged to get final filtered unique papers. We choose all of them individually as the aim of filtering is to remove the not important papers instead of selecting the important ones. For example, if a paper has high betweenness centrality, i.e., it can be based on an inter-disciplinary field but has low closeness centrality, i.e., as not much work on the topic has been done. we will keep that paper for further evaluation. But if a paper has not been cited much implies low in-degree and low hub score, not referencing many important papers (papers with higher number of citations) will have low out-degree, low authority score and low closeness centrality and also does not play an important role in connecting different communities – low betweenness centrality; such papers are filtered out. Either their content is not good enough or if not then the venue of their publication is not the best one for the work.

*4.3.3 Title similarity score calculation and cc bc algorithms*

For title similarity, python nltk wordnet is being utilised. It is open source package in python language which is trained on WordNet, a lexical database for the English language. It groups English words into sets of synonyms called *synsets*, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts. This inter linking is used to compute the similarity between different words.

[https://wordnet.princeton.edu/wordnet/citing-wordnet – for how to cite wordnet]

We need to check similarity based on the context. This algorithm is proposed by Mihalcea et al. in the paper “Corpus-based and Knowledge-based Measures of Text Semantic Similarity” [https://www.aaai.org/Papers/AAAI/2006/AAAI06-123.pdf]*.*

In an attempt to implement this algorithm, we used the following procedure:

1. We POS (part-of-speech) tag the sentence to tell the wordnet what POS we are looking for.
2. Since wordnet only contains information on nouns, verbs, adjectives, and adverbs. We ignore everything else.
3. Form synonym sets (Synset) for each of word in both the sentences.
4. For each word in the first sentence,
   1. Get similarity value of the most similar word in the other sentence.
   2. Sum this value for all the words in the sentence.
5. Average the sum value by dividing with the number of words. Thus, giving the sentence similarity score for sentence 1 with sentence 2.

***Pseudocode:***

*Function:* Sentence\_similarity (sentence1, snetence2)

POS\_tag(sentence1), POS\_tag(sentence2)

Synset1 = get\_synset(sentence1), synset2 = get\_synset(sentence2)

For each word w1 in synset1

Best\_score = max (similarity score of w1 with each of w2 in synset2)

Final\_score = Best\_score / number\_of\_words(synset1)

Return Final\_score.

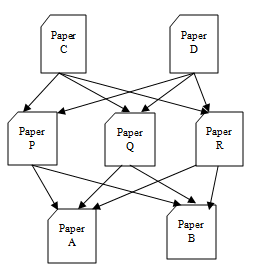
There are known problems with this implementation:

* The implementation is not exactly as in the paper since the max similarity is not weighted with an inverse-document-frequency.
* Wordnet has some issues with computing similarities between adjectives and adverbs

Title similarity for all the extracted papers (set-I and set-II) is calculated. Top from set-I are chosen by selecting the papers whose title similarity score is higher than the average title similarity of all the papers in set-I. From set-II only the top 10 (depending on the number of required journals this value can be varied) papers are selected even when they don’t have references they are showing very high similarity with our seed paper and hence their journals can be useful. (And if only the title had matched it will get discarded in the forthcoming steps)

*Bibliographic coupling (BC) and co-citation (CC) score*

Bibliographic coupling is a measure that is the number of pairs of papers that refer the same paper, whereas co-citation is the number of pairs of papers that are cited together by other papers [29].



**Fig 4. Example of bibliographic coupling and co-citation analysis**

In figure 4, papers C and D both reference papers P, Q, and R. Hence, their bibliographic coupling score is 3, 3 and 3 each. It can be calculated as follows:



B. C (Ck, Dk) is one if papers C and D cite paper k.

Papers A and B, are both cited by papers P, Q, and R, forming a co-citation. Thus, the co-citation score of A nd B is 3 and 3. It can be calculated as follows:



C.C (Ak, Bk) is one if paper k cites papers A and B.

The purpose of B.C score and C.C score is not to analyse the indirect relationship between the papers in a multilevel network. Therefore, B.C strength and C.C strength of papers C and A are calculated independently.

We define the C-Score by merging of these two measures to reflect both characteristics:



In the C-score, the numerator represents the similarity of two papers based on citation information, and the denominator is the distance between the paper and the paper of interest on the network. Therefore, C-score can be considered to be a combination of citation analysis and network analysis. The numerator is the sum of the bibliographic coupling strength and co-citation strength of paper P. J represents all papers except paper P, which is a target of the C-score. The C-score measures how strongly P is related with other papers, J, in both aspects. Thus, a high value of this numerator is an indication that P has a related subject matter with its neighbours. On the other hand, a low value indicates that P is not relevant to the contents of other papers. The C-scores consider the relevance of P with not only J, but also I. The denominator of the C-score determines the boundary of the research area, which has papers that are more relevant and closer to I. d (I, P) is the distance, considered to be the number of links between I and P. The more distance there is between them, the more indicative that the topic or domain field of the two papers is different.

*4.3.4 Link weightage algorithms and key-route search*

Main path analysis is a bibliometric method to trace the most significant path in a citation network. For identifying the main path in any network, the links in the network are given weights using traversal count. Traversal counts measure the importance of a link. In the next step, the most significant links are added to form the “main path”.

Types of traversal counts:

* Search path count (SPC):

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.

* Search path link count (SPLC):

A link’s SPLC is the number of times the link is traversed if one runs through all possible paths from all the ancestors of the tail node (including itself) to all the sinks.

* Search path node pair (SPNP):

A link’s SPNP is the number of times the link is traversed if one runs through all possible paths from all the ancestors of the tail node (including itself) to all the descendants of the head node (including itself).

In our research we consider the SPNP for weighing the links because

* SPC considers the path starting from the parent paper which can be very old to the latest paper which has just arrived and might not be referenced or viewed much. Also, the number of paths considered in by SPC is very general and the weights do not vary much between individual links.
* SPLC on the other hand solves the very old parent node problem but while giving weight to the current link it considers the significance of all its descendants. For example, paper B is referenced by paper A which is not a very good paper, but paper A is being referenced by another paper C, which is of very good quality and hence have a lot of citations. Due to the high importance of paper C, paper A links will be given high weightage. But paper A is not good itself.
* Hence to avoid this SPNP is chosen which considers only the ancestors and the descendants of the linked nodes.

The next step after performing the traversal counts is the *Key-route search***.** It is designed to choose the significant links in both the local and global search. Another advantage of the key-route approach is that one can control the detail of the main paths by varying the number of key-routes. Higher the number for key-route is specified, higher are the details revealed. A problem that the main path approach suffers is that the link with the highest traversal count may or may not be included in the main path. As a solution to this problem, the main path is viewed as an extension of the most significant link and the search begins from both the ends is the key-route rather than just from the sources.

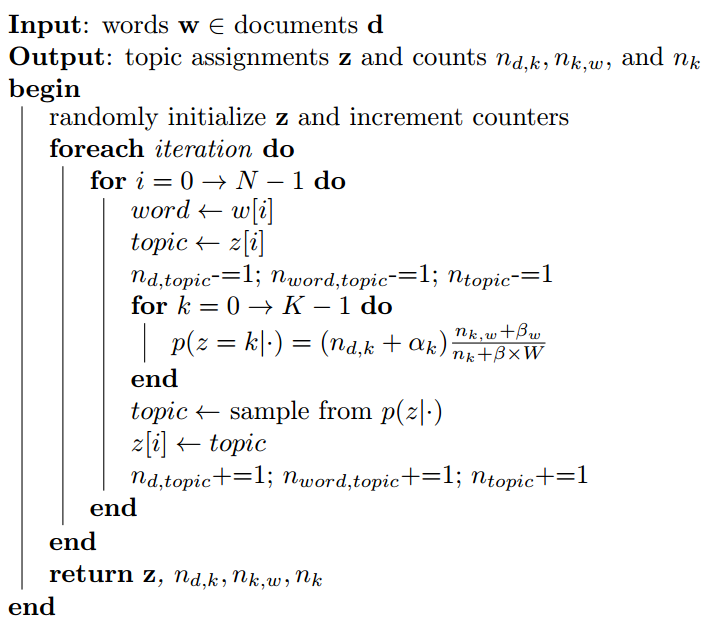
We call this the key-route search. It guarantees that this key route is included in the main path. The key-route search procedure is as follows.

* Select the key-route; it is the link that has the highest traversal count.
* Search forward from the end node of the key-route until a sink is hit.
* Search backward from the start node of the key-route until a source is hit.

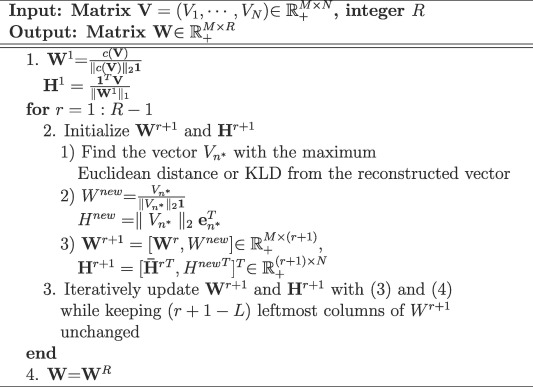
The search in Steps 2 and 3 can be either local or global. One also can select multiple key-routes and execute the procedure multiple times, each time selecting the link with the next-highest traversal count, to obtain multiple key-route main paths.

*4.3.5 LDA and NMF algorithm and how it works*

Each document, in our case each abstract can be viewed as a mixture of topics. Latent Dirichlet allocation (LDA) is an example of a topic model. It assigns a set of topics to the abstract with certain probability. The number of topics to be divided needs to be can be specified.

**

Non-negative matrix factorization(NMF) is a factorization model which factorizes the document term matrix in 2 smaller matrices. Document topic matrix and topic term matrix. This again gives us a set of topics with a confidence score.



We use these methods to find out similar abstracts to the given paper by running the algorithms for all the shortlisted paper abstracts and the seed paper’s abstract. The topic with maximum score in the seed paper is selected as the main topic and all other papers are ranked based on their scores of that topic. This ranking is done separately for LDA and NMF as LDA mainly considers the terms as individual in the documents to decide the topics and hence uses term frequency vector as the input where as NMF tries to capture the words occurring together in one topic using the tfidf vector.

*4.3.6 Ranking of the similar abstracts*

The 2 ranked lists of the abstracts are merged based to get the best of both the algorithms. Here we use the rank of the paper in list-I (LDA list) multiply with the topic score and add it with the product of the rank of the same paper in list-II (NMF list) and its topic confidence to get a new score. These scores are calculated for all the papers and then sorted in increasing order. Unique venues are finally selected based on these papers in the same order.

**5. EXPERIMENTS**

**5.1 Data Description**

The Microsoft Academic Graph can be accessed via the Microsoft Cognitive Services Academic Knowledge API. It is currently being updated on a weekly basis. The data is a heterogeneous graph that has information relating to academic papers and their venues. With the service of Microsoft Cognitive Services Academic Knowledge API, it is possible to process user queries for academic content and retrieve the information. It models scholarly communication activities which consists of six types of entities publications, authors, institutions (affiliations), venues (journals and conferences), fields of study and events (specific conference instances); and the relations between these entities citations, authorship, etc. The dataset also contains metadata of the papers, such as year of publication, title and DOI. The MAG is the largest publicly available dataset of scholarly publications and the largest dataset of open citation data. It also has a very good coverage across different domains with some bias towards technical disciplines. On the other hand, its limited to completeness as only 30 million papers out of 127 million have citation data. Similarly, as with the author and affiliation entities, the papers in MAG are linked to publication venues like journals and conferences.

*5.1.1 Datasets Construction.* We constructed the dataset by crawling data from Microsoft Cognitive Services Academic Knowledge API. We further processed the dataset by retaining only papers related to field of study of computer science.

Information about which field or fields of study does a publication belong to is very useful. Also, this information is often very difficult to get, as it is dependent on either having access to the content of the publication or to the manually created metadata. An investigation on the fields of study provided by MAG for papers to understand the reach of the dataset. All the fields of studies found in MAG are organised hierarchically into four levels (level 0 to level 3, where level 3 has the highest granularity). There are 47,989 fields of study at level 3 (e.g. "COBOL"), 1,966 at level 2 (e.g. "Low-level programming language"), 293 at level 1 (e.g. "programming language") and 18 at level 0 (e.g. "computer science"). Out of the 126,909,021 total papers, 41,739,531 (about ~ 33%) are linked to one or more field of study entities. One part of the dataset which is extensively used in this paper is the citation network. The MAG contains 528,682,289 internal citations (citations between the papers in the graph). This means each paper in the graph is cited on average 4.17 times. However, a significant portion of the papers are disconnected from the network (neither cite nor are cited by any other papers). There are over 80 million such nodes.

However, in spite of the limitations, the MAG is currently the most comprehensive publicly available dataset of its kind and represents a considerable effort which proves useful in many areas of research. For our study we used the version of MAG published on 5 February 2016.We generated a subset of the papers in the dataset based on their field of study. Computer science has been taken as a main field of study to build the venue recommender system. Information Retrieval, Machine Learning <add all the topics for experiment here> were the some fields of study that we considered for building our citation network to recommend venue to researcher. The major attributes of the MAG datasets are mentioned in Table 1.

***Table 1. Microsoft Academic Graph (MAG) Dataset***

|  |  |
| --- | --- |
| Papers | 126,909,021 |
| Authors | 114,698,044 |
| Institutions | 19,843 |
| Journals | 23,404 |
| Conferences | 1,283 |
| Conferences Instances | 50,202 |
| Fields of Study | 50,266 |

**5.2 Experimental Settings**

Our experiments are conceived based on a scenario in which the user has a research title on mind along with the features like keywords, field of study and abstract of the written paper and is searching for a suitable venue (journal/ conference). The driving motivation behind our work is to investigate the possibility of replacing existing academic recommender system to recommend more relevant venues. To comprehensively evaluate our proposed method, more specifically we address the following research questions (RQ):

* RQ1: How does SNAVER perform in handling cold-start problem in case of a new researcher?
* RQ2: What are the effects of keywords (number and related keywords with the query topic) with the end results of the proposed personalized venue recommendation?
* RQ3: How does our SNAVER approach perform as compared with other state-of-the-art venue recommendation methods?
* RQ4: Does SNAVER consistently outperform other existing algorithms irrespective of domain with respect to available information?
* RQ5: What are the effects of different hyper-parameter settings? (e.g., Key route selection, number of topics for LDA and NMF, combined similarity measure information for personalized venue recommendation?
* RQ 6: What is the performance of our final integrated recommender system for the task of personalized venue recommendation?

*5.2.1 Baseline.* For the purpose of demonstrating the effectiveness of our proposed approach, we compare our results across several baseline algorithms:

* ***Elsevier Journal Finder:***

To evaluate the accuracy of our proposed system, we applied a strategy of comparing the NDCG values of our approach with Elsevier Journal Finder results. We randomly selected 100 topics from various domains of computer science field which needs to be extracted from the testing dataset. The target researcher has to give the following parameters as input.

* *Input:* Paper title, Paper abstract, Field of research
* *Output:* A maximum of ten journals

After giving the title and abstract as input to the Elsevier journal finder we have retrieved the results. It is known the Elsevier journal finder can suggest up to a maximum of ten journals as relevant venues.

* ***Springer Journal Suggester:***

Also, the same topics which were tested in Elsevier journal finder were tested with Springer journal suggester. After giving the title and abstract as input to the Springer journal suggester we have retrieved the results. As known, the Springer journal suggester can suggest up to a maximum of twenty journals as relevant venues.

* *Input:* Paper title, Paper abstract, Field of research
* *Output:* A maximum of twenty journals

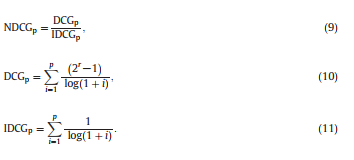
*5.2.2 Comparison with other approaches*

* ***Simple Counting:***For each target paper pi, we simply count the occurring frequency of venues of three kinds of neighboring papers of paper pi, i.e., the reference neighboring papers (papers cited by pi, referred as Simple Count-Ref), sibling neighboring papers (papers that share at least on citation with pi, referred as Simple Count-Sibling) and author neighboring papers (other papers written by authors of pi, referred as Simple Count-Author).We also count the frequency of venues of the combination of all three kinds of neighboring papers (referred as Simple Count-All). We would then rank and return the venues in terms of their frequency.
* ***Content-based LDA:***We construct a profile for each venue by concatenating all the papers published in it. We use LDA topic model implemented by Mallet [15] to retrieve the topic distribution for each paper and venue over 100 topics. We then compute and rank venues by their similarities with the target paper.

**5.3 Evaluation Metrics**

We employed four metrics, precision, recall, F1 score and NDCG, to evaluate the performance of SNAVER. Detailed information about these metrics has been discussed. All experiments we performed on a 64-bit and 2.4GHz intel core i5, 8-G bytes memory, and implemented with python 2.7.1.

From the recommender system literature, we learn that performance evaluation is mostly conducted by using fewer than 25 items because recommending too many items can confuse users [39,40]. Hundreds of researchers with expertise in the subjects of the papers provided their recommendations. The titles, authors, year of publication, and venue names of the recommended papers are provided to the experts. The experts determine whether they are satisfied or not with recommended venues. For evaluation, normalized discounted cumulative gain (NDCG) and mean reciprocal rank (MRR) are used [41,42]. NDCG measures the performance of a recommender system based on the graded relevance of the recommended items.



NDCGp represents the total normalized gain accumulated at a particular rank p. DCGp is the total gain accumulated at a particular rank p. The relevance value, r, of recommended item is tertiary ; r ∈ {0, 1, 2}. It is set to two if the user agrees that the research paper’s scope and the venue scope match, set to one if there is partial matching or it is set to zero otherwise. IDCGp generates the maximum possible DCG until rank p for normalization. All NDCG calculations are then relative values on the interval 0.0 to 1.0. In a perfect recommendation, the NDCG value is one because DCGp will be the same as IDCGp. Mean reciprocal rank (MRR) is widely used in the study of information retrieval and measures the ability of a recommender system to return a relevant item at the top of the ranking.

To measure the recommendations’ performance, we measured precision, recall, and normalized discounted cumulative gain (NDCG) (Järvelin&Kekäläinen, 2002; McNee, Riedl, &Konstan, 2006). Precision is derived by dividing the number of relevant venues recommended according to the researcher’s interests by the number of recommended venues, as shown in Eq. Recall is derived by dividing the number of relevant venues recommended by the number of relevant venues, asshown in Eq. (8).

Precision = |relevant *v*enues ∩ top *v*enues|/ |top *v*enues| (8)

Recall = |relevant *v*enues ∩ top *v*enues| |relevant *v*enues| (9)

Discounted cumulative gain (DCG) measures the extent to which a venue ranking is relevant to a user’s ideal ranking, as shown in Eq. (10). Relevance value is the relevance assigned by a researcher to the venue at position p. We measured the normalized discounted cumulative gain (NDCG), as the ideal ranking, as shown in Eq. (10). As recommendation lists vary in length, we used NDCG. IDCGp is the maximum possible ideal DCG at position p.

NDCGp = DCGp/IDCGp (10)

**5.4 Performance Comparison**

We conducted experiments to evaluate the recommendation capabilities of the proposed SNAVER compared with Elsevier Journal Finder and Springer Journal Suggester. Elsevier Journal Finder and Springer Journal Suggester are chosen as comparison systems because they are the most recognized proprietary databases for journal content and provide recommendation services to customers [38].

We considered twenty sub domains of computer science field that were selected as papers of interest in our experiments. We chose multiple sub domain papers because we wanted to compare the recommendation results irrespective of the topic of the paper. Moreover, 100 papers are selected from various fields, including Information retrieval, image processing, security, wireless sensor network, machine learning, software engineering, computer vision, artificial intelligence, data mining, natural language processing, parallel and distributed systems, multimedia, world wide web, operating system, databases, programming languages, real time and embedded system, human-computer interaction, bioinformatics and computational biology. We performed a blind test to assess the validity of the proposed method. For each paper of interest, we compute the title similarity of all papers from the field of study computer science. Top selected papers whose similarity score are more than the average similarity score considered as candidates for further computation.

**5.5 study of the proposed approach**

The main findings with respect to our RQs are summarised below:

5.3.1 How does SNAVER perform in handling cold-start problem in case of a new researcher (RQ1)

* The analyses in Section 6 show that, in case of a new researcher also the proposed system will perform with the same level of accuracy while suggesting relevant venues. Because in this approach the co-authors records are not required while constructing the citation network. The proposed approach considering the titles matching with the query paper and extracting only relevant papers to generate the network for further computation. While the traditional approach has been taken the co-authors publications to draw the networks and applies random walk restart algorithm with a variation to suggest venues to the target researchers. But here the system will perform with the same relevance results irrespective of the types of researchers.

**5.3. 2** What are the effects of keywords (number and related keywords with the query topic) with the end results of the proposed personalized venue recommendation (RQ2)

* We have also observed the suggested venues for a same input with changing their input keywords and conclude that with less number of keywords or unrelated keywords the system may mislead the suggestion and result with less number of relevant venues. Sufficient number of keywords specific to the topic of the paper results with a significant number of relevant venues. The system requires a sufficient number of key words. Perception of related keywords and sufficient number of keywords may influence the system. So, the system may be influenced by the input information given by the user.

But to generalize greater than around 4-5 keywords specific to the topic of the paper is best.

**5.3.3** How does our SNAVER approach perform as compared with other state-of-the-art venue recommendation methods (RQ3)

* The proposed approach outperforms other state-of-the-art venue recommendation methods like Elsevier journal finder and Springer journal suggester etc. The analyses in Section 8 show that the proposed approach is efficient not only terms of in precision, recall and F1 score but also outperforming Elsevier journal finder in terms of NDCG @5 and @10. Similarly, it’s outperforming springer journal suggester in terms of NDCG @5, @10 and @20 respectively. So, it is possible to replace the existing venue recommender systems with more relevant suggestion of venues for target researcher.

**5.3.4** Does SNAVER consistently outperform other existing algorithms irrespective of domain with respect to available information (RQ4)

* The proposed approach outperforms other state-of-the-art venue recommendation methods like Elsevier journal finder and Springer journal suggester etc. Here we are using the huge dataset Microsoft Academic Graph (MAG). It contains various domains with sufficient number of papers to carry out our research in any discipline. As we have already tested the same after checking around hundred topics from around twenty sub domains of computer science. And we have observed that the proposed system is performing stability in terms of performance measures like precision, recall, F1 score and NDCG respectively. Overall the performance has been recorded as consistent irrespective of domains and topics.

**5.3.5** What are the effects of different hyper-parameter settings (e.g., Key route selection, LDA similarity measure, NMF similarity measure, combined similarity measure information for personalized venue recommendation (RQ5)

* We have tested the proposed system with all variations of key route selection. We have identified the main path analysis through various options like forward path, backward path and key route respectively. But observed that the key route is the only one method which is returning significant papers with reputed venues. So, tested all hundred topics with the key route main path analysis technique. We have also checked the abstract similarities with both LDA and Non-negative matrix factorization methods and observed that while merging it results in a better ranking order of venues. So, opted the integrated approach for ranking final venues before recommending relevant venues to target researchers.

5.3.6 What is the performance of our final integrated recommender system for the task of personalized venue recommendation (RQ6)

* The proposed approach outperforms other state-of-the-art venue recommendation methods like Elsevier journal finder and Springer journal suggester etc. The analyses in Section 8 show that the proposed approach is efficient not only in terms of precision, recall and F1 score but also outperforming Elsevier journal finder in terms of nDCG @5 and @10. Similarly, it’s outperforming springer journal suggester in terms of nDCG @5, @10 and @20 respectively. So, it is possible to replace the existing venue recommender systems with more relevant suggestion of venues for target researcher.