I J C T A, 9(3), 2016, pp. 1753-1763

Recommending Top Research Categories to Naive Researchers

Keerthi Krishnan*, and K.S. Easwarakumar**

Abstract: There is a tremendous increase in the amount of information available online since the development of digital media. Specifically, a large number of scientific research publications are available in emerging areas of different research domains. The amount of research articles published is increasing every year and also several technologies are available to digitize research articles that are not published online. In such a scenario, finding the current research areas or topics in a specific field of research is a major concern for new research scholars. In this paper, a recommendation system is proposed, which suggests top research categories to new researchers who are striving to identify recent scientific articles in their area of interest. This work will help budding researchers to identify a research direction.

Keywords: Social Networking, Recommendation System, Research Categories, Document Frequency based Clustering

1. INTRODUCTION

The growth of ongoing research particularly in Computer Science (CS) domain is enormous and most of them are digitally available in the web. To select research papers in specific research fields from this huge volume of publications, scholars need to do a keyword based search in various search engines, or browse top conference proceedings and various journal articles in their interested areas. Also, researchers can access several online archives like DBLP, Citeseerx etc, which are specifically for research articles in the CS domain. In case of budding researchers, this is a time consuming process, as they are not aware about the top conferences and good journals in their research field and also the technical terms to be used as keywords in various search engines such as google scholar and also various other online archives. Hence, there is a need for a recommendation system to suggest top research areas in a particular field so that these researchers can use them as search keywords for doing their search efficiently.

Recommender systems represent state-of-the-art tools for efficient selection of the most relevant and reliable information resources, and the interest in such systems has increased dramatically over the last few years. Generally, recommender systems help users to identify products or services (such as books, electronic gadgets, movies, web pages etc), based on analyzing the preferences from users. Recommendation systems are proposed in different application domains such as book recommendation [17], recommendation system for tourism [6,19], movie recommendation [14,16], recommendation of tv-programs and other entertainments [20], etc.

Various recommendation systems related to e-learning are also proposed in [3]. A recommender system to determine the efficiency in learning of a research student by analyzing the student's behavioral and psychological aspects is proposed in [13]. A course recommendation system is proposed in [1], which recommends the best combination of courses to distance learning students in the engineering education by

* Research Scholar, Department of ComputerScience & Engineering, Anna University, Chennai 600 025, INDIA, *Email: keerthirsau@gmail.com*

** Professor, Department of ComputerScience & EngineeringAnna University, Chennai 600 025, INDIA, *Email: easwarakumarks@gmail.com*

collecting data regarding student choice of courses using moodle. A detailed analysis of the entire CS research is given in [11] by analyzing articles published in past two decades. An attempt to investigate how the researchers change their research fields is discussed in [4] by analysing the scientific career of researchers in computer science domain. A document centric approach to identify researchers with multiple expertise is proposed in [18].

Several studies related to recommending research papers are also available in the literature. Research paper recommender systems was introduced in 1988 as part of the citeseer project [7]. Several search engines such as *citeulike*, *Docear*, etc are available in the web, which recommends research papers. A detailed survey on research paper recommender systems is given in [2]. A trust based recommender system for scientific publications is proposed in [10] for recommending research papers to readers based on the opinion of a trust worthy person who is an expert in the research domain of the specific research paper.

In this paper, a Research Category Recommendation System (*RCRS*) is proposed, which recommends the top research categories to the new researchers who are striving to identify recent research topics in their research field. Thus the new researchers can do their literature search efficiently by giving these topics as their search keywords and can save a lot of time. To the best of our knowledge, no recommendation systems are proposed for suggesting research categories or research areas to naive researchers.

Rest of the paper is organized as follows. Detailed explanation of the proposed recommendation system is given in section 2. Description of the dataset used and result analysis are discussed in section 3. Finally, section 4 deals with the concluding remarks and future directions.

2. PROPOSED RECOMMENDATION SYSTEM

In this paper, a recommendation system is proposed to suggest the top r -research categories in a particular field to budding researchers. The two parts of a typical recommender system are users and items. In this work, users refer to new researchers and items indicate top research categories. Focus of this work is mainly in the CS domain, specifically in the Computational Geometry (CG) field. Full content of research articles published in this field are considered for evaluation. Since research articles are considered as documents, the words ‘document’ and ‘article’ are used interchangeably.

2.1. Preprocessing

Preprocessing steps such as tokenization, stopword removal and stemming are needed to represent the set of documents in the term frequency based matrix representation. Tokenization converts the document into set of distinct words. Stopword removal eliminates words having no meaning such as *a*, *as*, *the* etc. In this work, words such as *corresponding author, all rights reserved, email* and complexity related terms such as *lg*, *lglgn*, *nlglgn*, etc. are considered as special stopwords and are also eliminated. Stemming is done to convert words to their root forms. For example, *networks* and *networking* are converted to their root form *network*. Similarly, *spanning* is converted to *span* and hence the meaning of the keyword *spanning tree* is lost. In this work, such technical terminologies are given importance and hence customized stemming is done to convert words ending with ’s’ and ’*ies*’ to their root forms. For example, the term *directed graphs* is converted to *directed graph* and *range queries* is converted to *range query*. A keyword list is created by extracting author defined keywords written in between *keyword* section and *abstract* section of each article. This list is further refined by eliminating duplicate keywords and a distinct list of keywords is obtained.

2.2. Keyword Document Matrix Representation

Let $D = \{d_1, d_2, ..., d_n\}$ be the set of documents and $W = \{w_1, w_2, ..., w_m\}$ be the refined keyword set, where n is the total number of documents and m is the total number of keywords. Each document in the collection is represented as vector of dimension m. Frequency of each keyword is considered as its weight, which means that keywords appears more frequently in a document are given more importance. Let $N(w,d)$ be the frequency of keyword w in document d, then $N(w,d)$ can be defined formally as

$$N(w,d) = \begin{cases} 0 & \text{if } w \text{ is not present in d} \\ |w|_d & \text{otherwise} \end{cases}$$

where $|w|_d$ is the number of times w present in d. The vector representation of document d is,

$$\vec{d} = \left(N\left(w_1,d\right), N\left(w_2,d\right), ..., N\left(w_m,d\right)\right)$$

Thus, n such vectors are formed which are together represented as a Keyword Document Matrix (*KDM*) with m rows and n columns, where rows correspond to keywords and columns correspond to documents. Each research article generally consists of a very less number of keywords, hence *KDM* is very sparse.

KDM can be visualized as a bipartite network [5] in which two kinds of nodes co-exist with links only between nodes of different kinds as shown in figure 1. An undirected bipartite graph can be represented as a triple $BG = (D, W, E)$, where D and W are the two set of vertices such that $D \cap W = \varnothing$ and E is the set of edges in the graph. Let f be a function represents the relationship between W and D. Formally, f can be defined as $f : W \times D \rightarrow \{0, 1\}$, where

$$f(w_i, d_j) = \begin{cases} 1 & \text{if } w_i \text{ is in } d_j \\ 0 & \text{other wise} \end{cases}$$

for $w_i \in W$ and $d_j \in D$. An edge between d_j and w_i indicates the presence of keyword w_i in the document d_j. No edge exists between documents or keywords as the graph is bipartite.

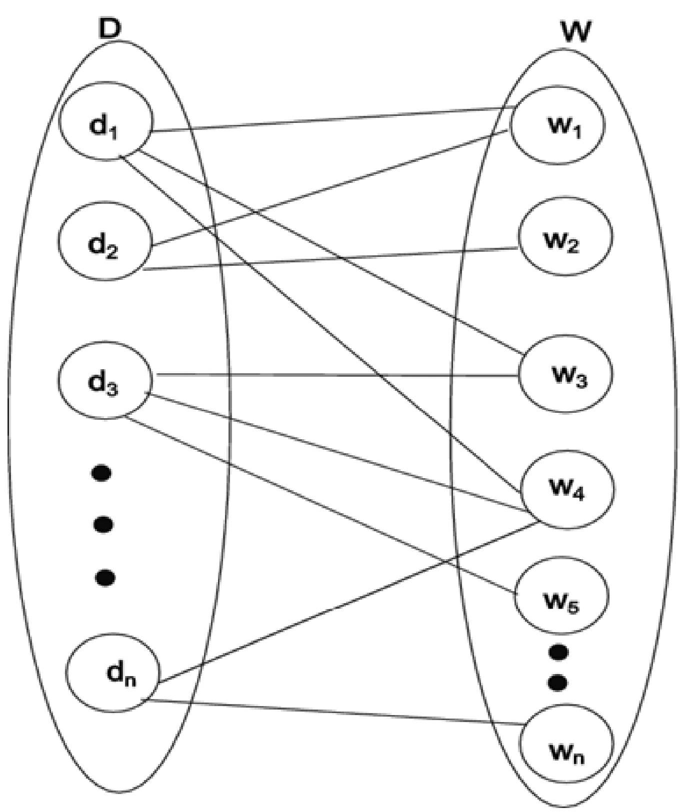

Figure 1: Bipartite graph representation of Keyword Document Matrix.

2.3. Document Similarity

For grouping the documents into several clusters, we need to compute the similarity among those documents. Generally, similarity measure gives a numerical value, which indicates the similarity between a pair of objects. In case of research articles, two documents are said to be similar, if they speaks about the same topic or they belongs to the same category. Various similarity measures for text documents are analyzed and evaluated with different datasets in [12] and the evaluation results indicates that Jaccard similarity coefficient performs well on research papers dataset. Jaccard coefficient (also referred to as Tanimoto coefficient) is used in this work to compute the similarity between research articles. Let $\vec{d}_a$ and $\vec{d}_b$ be the document vector of documents d_a and d_b. The Jaccard similarity of two document vectors $\vec{d}_a$ and $\vec{d}_b$ is

$$Jsim(\vec{d}_a, \vec{d}_b) = \frac{\vec{d}_a \cdot \vec{d}_b}{|\vec{d}_a|^2 + |\vec{d}_b|^2 - \vec{d}_a \cdot \vec{d}_b} \tag{1}$$

The Jaccard similarity co-efficient varies from 0 to 1. A value of 1 indicates that the two documents are same and 0 indicates that they are totally different.

$$Jsim(\vec{d}_a, \vec{d}_b) = \begin{cases} 1 & \text{if } \vec{d}_a \text{ and } \vec{d}_b \text{ are same} \\ 0 & \text{otherwise} \end{cases}$$

2.4. DOCUMENT FREQUENCY BASED CLUSTERING (DFC)

Clustering is a data mining task to divide a set of objects into several clusters such that objects in the same cluster are similar to each other than the objects in other clusters. Detailed explanations of various clustering techniques such as partitioning and hierarchical are given in [8]. In this work, clustering is applied to divide *D* into clusters of similar documents. The label assigned to each cluster indicates the research category of documents in that cluster. Instead of filtering documents belonging to a specific research category from the large set of documents, they can be retrieved from the cluster labelled by that category. Thus the search space can be reduced and hence response time is improved. We present a partitioning clustering algorithm, named as *Document Frequency based Clustering(DFC)*, which is the variant of K-means clustering algorithm [9] to group documents into *k* clusters.

Background: For a given given set of *D* documents and *k* number of clusters, the traditional K-means clustering algorithm work as follows. First, it randomly choose *k* documents from *D* as the initial cluster centroids and the remaining documents are assigned to any of the k clusters based on the similarity between documents and cluster centroids. Document *j* is assigned to cluster *x* if document vector $\vec{d}_j$ is more similar to centroid vector $\vec{c}_x$. Then, new cluster centers are computed by calculating the mean value of objects in each cluster and all the documents in *D* are re-assigned to *k* clusters based on the new centroids. This step is repeated until successive iterations give the same set of clusters.

In DFC, instead of selecting randomly, *k* initial cluster centroids (CC) are chosen as follows: The set of documents *D* is partition into *k* subsets D_1, D_2, D_k, each of size *p*, where $p = m/k$ m, is total number of documents and *k* is number of clusters. Each D_x $1 \le x \le k$, contains set of documents from l_{th} position till u_{th} position in *D*, which can be formally written as:

$$D_x = \bigcup_{l \le j \le u} \{d_j\}, \textit{ where } l = (x-1) \times p + 1 \textit{ and } u = x \times p$$

For example, if m = 250 and k = 10, then $D_1 = \{d_1, d_2, ..., d_{25}\}$, $D_2 = \{d_{26}, d_{27}, ..., d_{50}\}$, ..., $D_{10} = \{d_{226}, d_{227}, ..., d_{250}\}$. Centroid vector of x^{th} cluster is computed as $\vec{c}_x = \left(\left|d_j\right|_{w_i}\right)$, where $\left|d_j\right|_{w_i} = \left|\left\{d_j \in D_x : w_i \in d_j\right\}\right|$ is the number of documents in D_x which contains $w_i \in W$. The obtained k centroid vectors are stored in a set CC:

$$CC = \bigcup_{1 \le x \le k} \{\vec{c}_x\}$$

A formal description of finding initial cluster centroids is given in Algorithm 1.

Algorithm 1: Find Initial Cluster Centroids(D, K)

Input: D-set of documents and k-no of clusters

Output: set of initial cluster centroids

1. Initialize a set (CC) to store the centroid vectors of all clusters
2. $m \leftarrow |D|$;
3. $p \leftarrow \lfloor m/k \rfloor$;
4. foreach x, $1 \le x \le k$ do
5. $l \leftarrow (x-1) \times p + 1$;
6. $u \leftarrow x \times p$;
7. $D_x \leftarrow \bigcup_{l \le j \le u} \{d_j\}$;
8. foreach $w_i \in W$ do
9. $f_i \leftarrow \left|\left\{d_j \in D_x : w_i \in d_j\right\}\right|$;
10. endfor
11. $\vec{c}_x \leftarrow (f_1, f_2, \ldots, f_m)$;
12. $CC \leftarrow CC \cup \{\vec{c}_x\}$;
13. endfor
14. Return CC;

The similarity between each document vector $\vec{d}_j, 1 \le j \le n$ and the centroid vectors $(CC)\, \vec{c}_x, 1 \le x \le k$ obtained from Algorithm 1, is computed using Equation 1 and are stored in the similarity matrix DS. Based on the similarity values, each document d_j is put in the cluster c_x, where x corresponds to the column index of the most similar centroid, which can be determined as:

$$x = \left[\max_{1 \le x \le k} DS(j, x)\right]_x$$

For each such newly created C_x, new centroid vectors are computed as

$$\vec{c}_x = \left(\left| d_j \right|_{w_i} \right), \text{ where } \left| d_j \right|_{w_i} = \left| \left\{ d_j \in C_x : w_i \in d_j \right\} \right|$$

is the number of documents in C_x which contains $w_i \in W$. The whole process is repeated until new set of clusters and old set of clusters are same. Formal description for finding set of clusters is given in Algorithm 2.

In the traditional K-means algorithm, the initial centroids are chosen randomly. But, in the proposed clustering algorithm, the set of documents is partitioned into k subsets and the frequency of keywords in documents under each partition is considered as the initial cluster centroids. In K-means Clustering algorithm, the centroid vectors are updated by calculating the mean value of objects in each cluster. In this work, documents are the objects, which are represented in the form of a vector and hence, taking the mean of document vectors in each cluster is not appropriate. So, for each cluster, the centroid vector is updated using the document frequency vector computed from the respective cluster. Documents which are more significant are grouped and the algorithm converges within a less number of iterations, compared to K-means algorithm.

Algorithm 2: Find clusters (D, W, k)

Input: D-set of documents, W-set of keywords,

k-no of clusters

Output: set of k clusters

1. Initialize a similarity matrix (DS) to store the similarity between each document vector and cluster centroid vector
2. $C_x^{new} \leftarrow \emptyset$, $1 \le x \le k$;
3. repeat
4. $C_x^{old} = C_x^{new}$, $1 \le x \le k$;
5. foreach $\vec{d}_j$, $1 \le j \le n$ do
6. foreach $\vec{c}_x \in CC$ do
7. $DS(j,x) \leftarrow Jsim(\vec{d}_j, \vec{c}_x)$;
8. endfor
9. endfor
10. foreach $\vec{d}_j \in D$ do
11. $x \leftarrow \left[\max_{1 \le x \le k} DS(j,x) \right]_x$;
12. $C_x^{new} \leftarrow C_x^{new} \cup \left\{ \vec{d}_j \right\}$;
13. endfor
14. foreach C_x^{new}, $1 \le x \le k$ do
15. foreach $w_i \in W$ do

$f_i \leftarrow \left|\left\{d_j \in C_x^{new} : w_i \in d_j\right\}\right|$;

endfor

$\vec{c}_x \leftarrow \left(f_1, f_2, \ldots, f_m\right)$;

$CC \leftarrow CC \cup \left\{\vec{c}_x\right\}$;

endfor

until $C_x^{old} = C_x^{new}$, $1 \le x \le k$;

The running time of K-means algorithm is O(knmq), where k is the number of clusters, n is the number of points, m is the dimension of each point, and q is the number of iterations. The proposed DFC algorithm is also depends on the above factors. But the way in which the clusters are created is different and accordingly the number of iterations will become less. Hence, asymptotically both algorithms have same complexity.

2.5. CG Taxonomy

A taxonomy for the CG (Computational Geometry) field is created containing various research categories in CG using the keywords retrieved from the research articles. Let T_t be the set of keywords grouped under the category t and n_c be the number of categories, then the taxonomy can be represented as a set $T = \{T_t : 1 \le t \le n_c\}$, where each T_t is labelled as 'cat_t'. Some of the categories listed in the constructed taxonomy are *Fundamental geometric problems, Geometric searching problems, Geometric algorithms, Discrete mathematics, Discrete geometry, Visibility problems, Data structures, Geometric modelling and topology, Applications*. Some keywords come under the category *Visibility problems* are *visibility graph, visibility polygon, weak visibility, visibility, art gallery problem, splinegon, vertex guarding, wireless localization problem, watchman tour, visibility decomposition, visibility queries, approximate visibility*. Our proposed system will organize research articles based on the top level categories only.

2.6. Category Set Identification

For each cluster C_x obtained as a result of *DFC*, number of documents belongs to category t, $1 \le t \le n_c$, is determined as

$$c\#_t^x = \left|\left\{d_j : d_j \in C_x,\ w_i \in d_j \ and\ w_i \in T_t\right\}\right|$$

The category set of each cluster contains category labels whose document count falls above a threshold value h. For each cluster C_x, the Category Set (CS) is determined as

$$CS_x = \bigcup_{1 \le t \le n_c} \left\{cat_t : c\#_t^x \ge h\right\}$$

2.7. Research Categories Recommendation

For each category t, $1 \le t \le n_c$, category name and its corresponding category count is defined as an ordered pair:

$$\left(cat_t, c\#_t\right) = \left(cat_t, \sum_{x=1}^{k} c\#_t^x\right)$$

Now, the categories are ordered based on the number of documents in each category. From this top-*r* categories are considered as the top research categories(TRC).

$$TRC_r = \bigcup_{1\le x\le k} \{cat_t : c\#_t \ge c'\#\}$$

where $c'\#$ be the r^{th} largest value in $\{c\#_t : 1 \le t \le n_c\}$. The system will recommend TRC_r to naive researchers so that they can use those categories as search terms inorder to narrow down their search space.

3. EXPERIMENTAL ANALYSIS

As a usual practice, pre-assigned category labels are used as a baseline criteria for evaluating the quality of clusters. Since pre-assigned category labels are not available for research papers in CG field, manually assigned category labels which are created by a set of experts in this field are used for the evaluation. In such evaluations, the clustering algorithm is said to be good, if the cluster label matches with the manually assigned label of that cluster[12].

While implementing the proposed algorithm in the dataset, more than one research categories are obtained for each cluster. The dominating category (category assigned to majority documents in that cluster) among them is given as the label for that cluster. Also, more than one cluster can have the same cluster label, if the dominating category of those clusters are same.

3.1. Dataset

Research papers published by Elsevier in Computational Geometry-Theory and Applications Journal [21] are taken to evaluate the proposed recommendation system. Since, our recommendation system suggests top research categories to budding researchers, research articles published from the year 2011 till 2015 are considered for analysis. In this time span, 300 articles were published in various volumes (ranging from vol. no. 44 to vol. no. 49) in the above mentioned journal. For manual verification, these articles are only considered. Research articles in *PDF* format are converted to plain text for processing. In this work, full content of the article from *keyword* section till *conclusion* section is considered. Pre-processing is done on each plain text to represent it in the form of a document vector.

3.2. Evaluation

The label of each cluster is verified based on the label assigned manually. For verification, the dominating category label of each cluster is considered. The quality of the proposed algorithm is evaluated using the evaluation measures such as *Accuracy* and *F-measure*. Detailed explanation of these measures are given in [15]. Here, manually assigned label is taken as *relevant category* and label assigned by the algorithm is considered as the *assigned category*. Let *RC* be the set of documents in the relevant category and *AC* be the set of documents in the assigned category, then

$$\Pr ecision = \frac{|RC \cap AC|}{|AC|} \tag{2}$$

$$Recall = \frac{|RC \cap AC|}{|RC|} \tag{3}$$

and

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

3.3. Results and Discussion

The proposed algorithm assigns more than one categories for each cluster and they are ordered based on the number of documents assigned to each one. The dominating categories in each cluster is shown in Figure 2.

Table 1 shows the result obtained for the various evaluation measures discussed in Section 3.2. These results are obtained by comparing categories assigned by the algorithm with the manually assigned one. The recall values of most clusters are 1.00 which means that the category labels matches with the actual ones. The average accuracy and F-score obtained are respectively 82% and 0.90, which indicates that the performance of the proposed algorithm is good.

Figure 3 shows the top research categories. Among them top are recommended as the top research categories in the CG research field. The top categories (= 5) are *Discrete Mathematics*, *Fundamental*

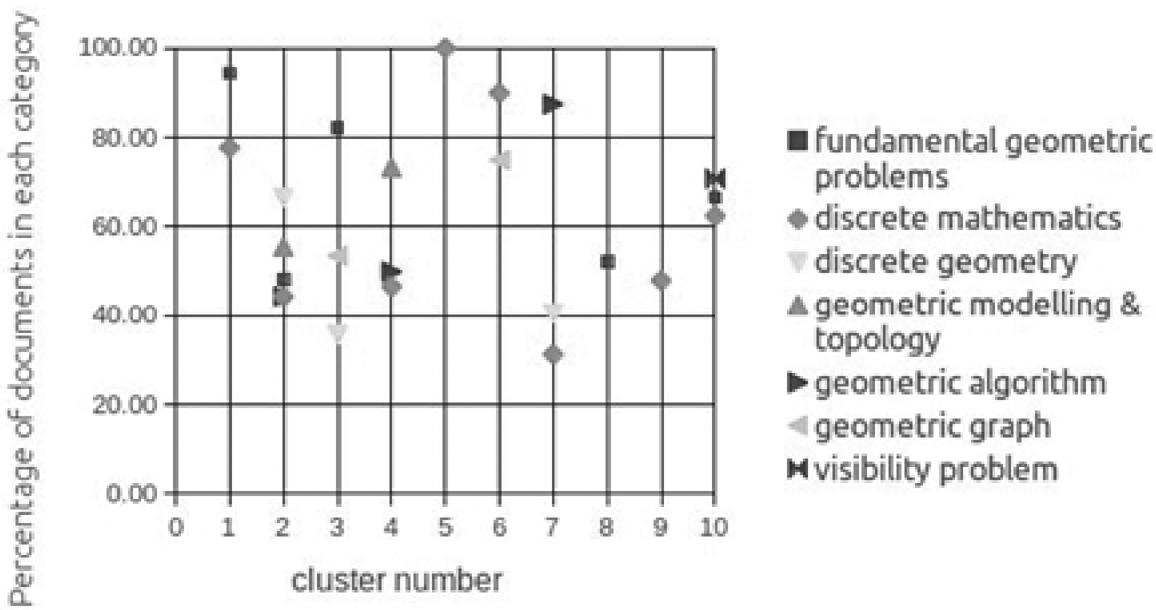

Figure 2: Dominating categories in each cluster

Table 1
Evaluation Results

Cluster No.	*Precision*	*Recall*	*F measure*	*Accuracy (%)*
1	0.82	1.00	0.90	83
2	0.83	0.95	0.89	78
3	0.84	1.00	0.92	86
4	0.78	1.00	0.86	73
5	1.00	1.00	1.00	100
6	0.83	1.00	0.91	85
7	0.72	1.00	0.84	75
8	0.77	0.95	0.85	74
9	0.80	0.94	0.87	80
10	0.83	1.00	0.91	85

geometric problems, *geometric algorithm*, *geometric graphs* and *discrete geometry*. The proposed system recommends these categories to new scholars who are interested to do their research in CG. These recommended categories are validated with the opinion of experts in the CG field.

To the best of our knowledge, there are no existing works for recommending top research categories. However, several research paper recommender systems are available for identifying the research papers[2]. Since this work on identifying the top research categories in a research field is a novelty, a direct comparison with the existing systems may not be applicable.

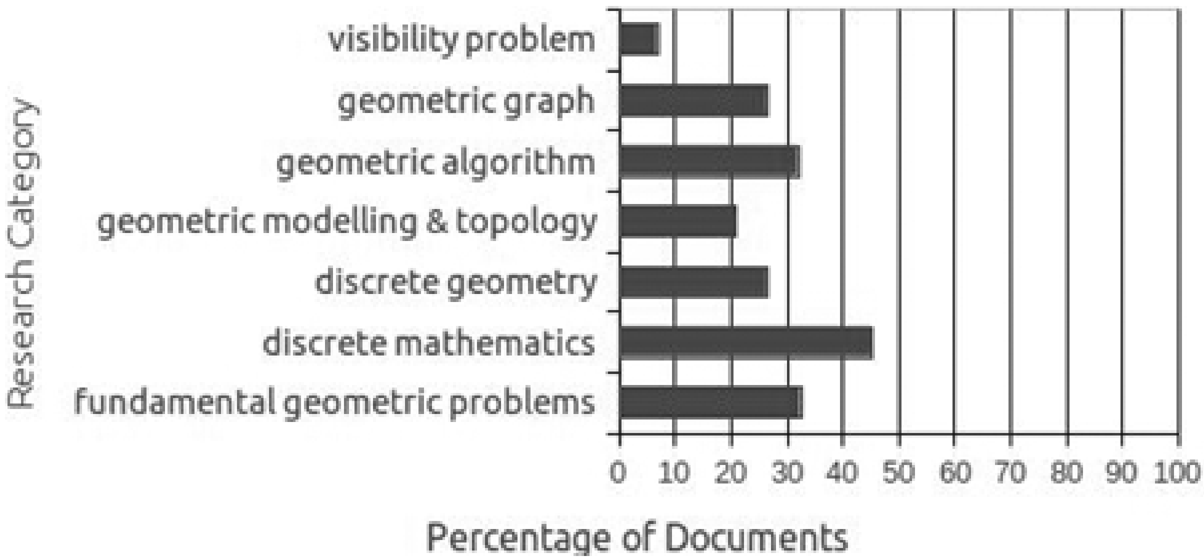

Figure 3: Top Research Categories

4. CONCLUSION

Top research categories in a particular research field is identified in this work which are useful to naive researchers in that field to perform their literature search effectively. A clustering algorithm based on document frequency is proposed to classify research articles into various research categories. As a result, researchers can narrow down their search space and save a lot of time by discarding the un-necessary articles. As a future work, the extracted keyword list can be further refined by eliminating less frequently occurring keywords and to narrow down our procedure so that exact research topics in the top categories can be suggested. Also, size of the dataset will be extended by including more journal articles and well known conference proceedings in Computational Geometry field.

References

[1] S.B. Aher. EM & AA: An algorithm for predicting the course selection by student in e-learning using data mining techniques. Journal of the Institution of Engineers (India): Series B, 95(1):43–54, 2014.

[2] J. Beel, B. Gipp, S. Langer, and C. Breitinger. Research paper recommender systems: A literature survey. International Journal on Digital Libraries, pages 1–34, 2015.

[3] J. Bobadilla, F. Serradilla, A. Hernando, et al. Collaborative filtering adapted to recommender systems of e-learning. Knowledge-Based Systems, 22(4):261–265, 2009.

[4] T. Chakraborty, V. Tammana, N. Ganguly, and A. Mukherjee. Understanding and modeling diverse scientific careers of researchers. Journal of Informetrics, 9(1):69–78, 2015.

[5] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 269–274, 2001.

[6] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. Mobile recommender systems in tourism. Journal of Network and Computer Applications, 39:319–333, 2014.

[7] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In Proceedings of the 3rd ACM conference on Digital libraries, page 89-98, 1998.

[8] J. Han and M. Kamber. Data mining: concepts and techniques. Morgan Kaufmann San Francisco, Calif, USA, 2001.

[9] J.A. Hartigan and M.A. Wong. Algorithm as 136: A k-means clustering algorithm. Applied statistics, pages 100–108, 1979.

[10] C. Hess. Trust-based recommendations for publications: A multi-layer network approach. TCDL Bulletin, 2(2):190–201, 2006.

[11] A. Hoonlor, B.K. Szymanski, and M.J Zaki. Trends in computer science research. Communications of the ACM, 56(10): 74–83, 2013.

[12] A. Huang. Similarity measures for text document clustering. In Proceedings of the sixth New Zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, pages 49–56, 2008.

[13] A. Kaklauskas, E.K. Zavadskas, V. Trinkunas, L. Tupenaite, J. Cerkauskas, and P. Kazokaitis. Recommender system to research students' study efficiency. Procedia-Social and Behavioral Sciences, 51:980–984, 2012.

[14] K. Madadipouya. A location-based movie recommender system using collaborative filtering. arXiv preprint arXiv:1508.01696, 2015.

[15] C.D. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval. Cambridge university press Cambridge, 2008.

[16] P. Melville, R.J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In American Association for Artificial Intelligence (AAAI)/IAAI, pages 187–192, 2002.

[17] R.J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In Proceedings of the fifth ACM conference on Digital libraries, pages 195–204, 2000.

[18] S. Pushpa, S. Elias, K.S. Easwarakumar, and Z. Maamar. Indexing scholarly publications using telescopic vector trees to discover authors with multiple expertise. International Journal of Information Studies, 2(3):166–173, 2010.

[19] W. Yang and S. Hwang. itravel: A recommender system in mobile peer-to-peer environment. Journal of Systems and Software, 86(1):12–20, 2013.

[20] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. User modeling and user-adapted interaction, 16(1):63–82, 2006.

[21] Computational geometry theory and applications, elsevier publications. http://www.journals.elsevier.com/computational-geometry/.