A Frequent Keyword-set based Algorithm for Topic Modeling and Clustering of Research Papers

Kumar Shubhankar
Centre for Data Engineering
IIIT Hyderabad
Hyderabad, India

shubankar@students.iiit.ac.in

Aditya Pratap Singh
Centre for Data Engineering
IIIT Hyderabad
Hyderabad, India

aditya_pratap@students.iiit.ac.in

Vikram Pudi
Centre for Data Engineering
IIIT Hyderabad
Hyderabad, India

vikram@ iiit.ac.in

***Abstract* - In this paper we introduce a novel and efficient approach to detect topics in a large corpus of research papers. With rapidly growing size of academic literature, the problem of topic detection has become a very challenging task. We present a unique approach that uses closed frequent *keyword*-set to form topics. Our approach also provides a natural method to cluster the research papers into hierarchical, overlapping clusters using topic as similarity measure. To rank the research papers in the topic cluster, we devise a modified PageRank algorithm that assigns an authoritative score to each research paper by considering the sub-graph in which the research paper appears. We test our algorithms on the *DBLP* dataset and experimentally show that our algorithms are fast, effective and scalable.**

Keywords. Closed Frequent Keyword-set, Topic Detection, Graph Mining, Citation Network, Authoritative Score

I. INTRODUCTION

Thousands of research papers are published every year spanning various fields of research. These research papers cover broad topics like Data Mining, Artificial Intelligence, Information Extraction, etc., which can be further divided into several sub-topics, sub-sub-topics and so on. Classification of research papers into the broad topics is a trivial problem which is generally dealt manually. But more often than not, a researcher is more interested in literature related to specific topics within these broader topics. For example, a researcher might be more interested in knowing about topics like Data Visualization, Frequent Pattern Mining, etc., which are different sub-topics within the broader field of Data Mining. Topic discovery has recently attracted considerable research interest [13, 15, 10, 21, 22, 24, 19, 4, 18, 8, 17]. Also, the researchers are often interested in knowing the important research papers from their topics of interest.

In this paper, we propose a novel and efficient method for topic modeling and clustering of research papers. The keywords associated with a paper can be used to deduce the topics the paper deals with. Often these keywords are fed as meta-data by the authors in the paper itself. But this is not the case for all the research papers. We need to have a deterministic way of determining the topics of concern for all the research papers. Based on the intuition that a document is well summarized by its title and the title gives a fairly good high-level description of its content, we use the keywords present in the title of a paper to detect the topics. It is to be noted that we do not use abstract of the paper for extracting phrases as there are lots of irrelevant phrases in the abstract present as noise.

In the proposed approach, we form *closed frequent keyword-sets* by top-down dissociation of keywords from the phrases present in the titles of papers on a user-defined minimum support. All the research papers sharing a topic form a natural cluster. We also propose a time independent, modified iterative PageRank algorithm to rank the research papers in a topic cluster which assign an authoritative score to each paper based on the citation network. For a topic *T* we consider all the research papers containing that topic and the citation edges of these papers. We can determine the *important* research paper belonging to a given topic *T* by looking at the scores of all the research papers belonging to that topic. It is to be noted that a research paper could belong to a number of clusters forming hierarchical, overlapping clusters.

These algorithms have many applications like recommendation systems, finding landmark papers, trend analysis etc. We test our algorithms on the DBLP dataset. Our experiments produce a ranked set of research papers corresponding to each topic that on examination by field experts and based on our study match the prominent papers from the topics in the dataset.

II. RELATED WORK

Topic extraction from documents has been studied by a lot of researchers. Most work on topic modeling is statistics-based like the work by Christain Wartena and Rogier Brusse [23], which uses most frequent nouns, verbs and proper names as keywords and clusters them based on different similarity measures using the induced k-bisecting clustering algorithm. There has also been some NLP-based work on topic detection like the work by Elena Lloret [16]. Our work is based on dissociation of phrases into frequent *keyword-sets,* which as shown in section 7 is very fast and highly scalable.

Clustering documents based on frequent itemsets [2] has been studied in the algorithms FTC and HFTC [3] and the Apriori-based algorithm [14]. Both of these works consider the documents as bags of words and then find frequent itemsets. Thus, the semantic information present in the document is lost. We extract phrases from the titles of the research papers and derive frequent substrings as frequent *keyword-sets,* maintaining the underlying semantics. Another work that talks about the preservation of semantics while forming topics is by Zhou Chong *et al.* [6], where the authors consider a window within which they find the itemsets which are candidates for topics. But, within this window, the relative position of the words is considered insignificant. We, on the other hand, have considered only sequential patterns within a phrase. It is quite intuitive that if we do not consider the relative positions of the keywords, there might be cases where the semantics are lost.

We have used closed [20] frequent *keyword-set* as similarity measure rather than maximal frequent *keyword-set* as used by Ling Zhuang *et al.* [25] in their work on document clustering. We cannot use maximal frequent *keyword-sets* as topics because then most of the information is lost as it considers only the longest possible *keyword-set*.

Topic summarization and analysis on academic documents has been studied by Xueyu Geng and Jinlong Wang [9]. They have used LDA model to extract topics and Kullback-Leibler divergence as similarity measure for clustering the documents. The drawback of LDA model is that it needs a pre-specified number of latent topics and manual topic labeling. In our study, no *prior* knowledge of topics is required. The work in [11] uses the correlation between the distribution of terms representing a topic and the distribution of links in the citation graph among the documents containing these terms. We on the other hand, use frequent *keyword-sets* to form the topics and utilize the citation links to detect important topics among the topics derived.

Rest of the paper is organized as follows: Section 3 describes our topic modeling algorithm. Section 4 discusses our approach to cluster the research papers into topics. Section 5 describes our algorithm for ranking of research papers using citation network. Section 6 presents the experiments and results. Section 7 talks about some applications of our algorithms and section 8 concludes our work.

III. OUR APPROACH

The method proposed by us is based on the formation of *keyword-sets* from titles of the research papers and finding closed frequent *keyword-sets* to form the *topics*.

Definition 1. Phrase: *A phrase P is defined as a run of words between two stop-words in the title of a research paper.*

Definition 2. Keyword-set: *We define a keyword-set K as an n-gram substring of a phrase, where n is a positive number.*

Definition 3. Frequent Keyword-set: *A keyword-set K is said to be frequent if its count in the corpus is greater than or equal to a user-defined minimum support [1].*

Definition 4. Closed Frequent Keyword-set: *We define a closed frequent keyword-set as a frequent keyword-set none of whose supersets has the same cluster of research papers as it. Each closed frequent keyword-set represents a unique topic T.*

A. Phrase Extraction

Given the title of a research paper R_i, we extract all its *phrases* P_{ij}. As defined above, a *phrase* is a run of words between two stop-words from a comprehensive list of *671* Standard English stop-words. Each research paper is thus mapped to the corresponding phrases present in its title.

Our original problem domain is:

$$\mathbf{R_1 - [P_{11}, P_{12}, P_{13}, \ldots]}$$
$$\mathbf{R_2 - [P_{21}, P_{22}, P_{23}, \ldots]}$$
$$\ldots$$

Here, R_i represents the i^{th} research paper and P_{ij} represents its j^{th} phrase. Our next step is to derive *keyword*-sets from these phrases, dissociating each phrase into *keyword-sets*. It should be noted that in the original problem domain, we will have to dissociate a given phrase multiple times in a single scan as a phrase might occur in several research papers. So, we reverse map the problem domain, mapping each phrase P_i to the set of research papers R_{ij} it belongs to, in one scan of the dataset. In this domain, each phrase will be dissociated only once. The problem domain is thus modified as follows:

$$\mathbf{P_1 - [R_{11}, R_{12}, R_{13}, \ldots]}$$
$$\mathbf{P_2 - [R_{21}, R_{22}, R_{23}, \ldots]}$$
$$\ldots$$

B. Keyword-set Extraction

As defined above, a *keyword-set K* is basically a substring of a *phrase P*. In our method, we have considered only the substrings of the phrases as *keyword-sets* and hence the relative ordering is maintained, preserving the underlying semantic of the phrases. Each *keyword-set* thus formed is a semantic unit that can function as a basic building block of knowledge discovery and hence is a potential *topic.*

As an example of *keyword-set* extraction, consider the phrase *ABCDE,* the potential frequent *keyword-sets* are the set of all the ordered substrings. Thus, *ABCDE* gives the following *keyword-sets*, as shown in figure 1.

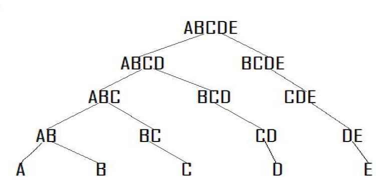

Figure 1. Set of all *keyword-sets* obtained from phrase *ABCDE*

Note that *ABCDE* is a phrase consisting of the keywords *A*, *B*, *C*, *D* and *E*. Finding all the substrings requires a simple implementation of queue in top-down fashion, taking *O(1)* time at each level and *O(n)* time overall. We consider only the substrings of a phrase rather than the power set of the keywords in the phrase. Thus, finding the *keyword-sets* requires *O(n)* time instead of $O(2^n)$.

C. Frequent Keyword-set Formation

Frequent keyword-sets are formed on a user-defined *minimum support*. *Frequent keyword-sets* are those *keyword-sets* whose support is no less than the minimum support. The supports of the *keyword-sets* are calculated during the generation of the *keyword-sets* from the *phrases* in the second scan.

It is to be noted that the length of the list of research papers corresponding to a phrase is its *support*, assuming that a phrase occurs not more than once in the title of the research paper. In the first scan, we cannot eliminate the phrases whose support is less than the minimum support as two or more phrases can share the same *keyword-set* whose combined support might be greater than the minimum support. The elimination of non-frequent *keyword-sets* would be done only after all the *keyword-sets*, along with their supports, have been generated in the second scan of the dataset.

The algorithm to increment the support and add research papers to a given *keyword-set* is shown below:

```
Procedure 1: Frequent Keyword-set Generation

Require: phraseKeys PK, minimum support min_sup

1: for each phrase P in PK
2:     keywordSetList KSL= findAllSubstringOf(P)

3:     for each keywordSet K in KSL
4:         keywordSetCount[K] += 1;
5:         add paper R to keywordSetPaperList[K]

6: for each keywordSet K in keywordSetCount
7:     if keywordSetCount[K] < min_sup
8:         delete(keywordSetCount[K])
9:         delete(keywordPaperList[K])
```

In the procedure 1, all the frequent *keyword-sets* are derived along with their supports. From *step 1* to *step 5*, all the *keyword-sets* of each phrase are derived and their supports in *keywordSetCount* and the corresponding paper list in *keywordSetPaperList* are updated. From *step 7* to *step 10*, we delete those *keyword-sets* whose supports are less than the *minimum support*. The *keyword-sets* thus derived are the frequent *keyword-sets.*

We have derived the frequent *keyword-sets* in a top-down fashion. As said before, we only need the substrings of the phrases and not the subsets, which would take $O(2^n)$ time. Traditional association rule mining algorithms like *Apriori* that require one scan of the dataset to calculate the supports of the itemsets at each level take too much time and space. In our algorithm, we require only *2* scans of the dataset to calculate the supports of all the candidate *keyword-sets*. Since our algorithm runs in *linear* time compared to *exponential* Apriori-like algorithms and takes only 2 scans of the dataset to calculate the supports, our algorithms are fast and highly scalable, as shown experimentally in section 6. Also, in Apriori-like algorithms which build higher length itemsets from smaller ones, the relative ordering between the itemsets is lost. In our method, we have considered only the substrings of the phrases as *keyword-set*s and hence the relative ordering is maintained preserving the underlying semantic of the phrases.

D. Closed Frequent Keyword-sets as Topics

At this point, we have the frequent *keyword-sets.* The papers that share the same *keyword-set* lie in the same cluster. In our algorithm, we may derive non-closed frequent *keyword-set*s as well. For example, *ABCD* and *ABC* may have the same list of papers in their clusters. In this case, we can remove *ABC* as it does not have any information that *ABCD* does not have. Our topic should consist of the maximal number of common keywords present in all the papers in the cluster. Thus we need to have *closed frequent keyword-sets* as topics. Notice that we calculated the support of the frequent *keyword-sets* in top-down fashion in a single scan in procedure 1. So we cannot eliminate the non-closed *keyword-sets* in procedure 1 itself.

To eliminate the non-closed frequent *keyword-sets*, we iterate level-wise in the list of *frequent keyword-sets*. We store the *frequent keyword-sets* in a level-wise manner, with the number of keywords in the *keyword-set* representing its level. For every *keyword-set* of length *i*, we iterate over the list of *keyword-sets* of length *(i+1)* and if *i*-length *keyword-set* is a substring of *(i+1)*-length *keyword-set* and the support is same for both, we delete the *i*-length *keyword-set,* as it is non-closed.

IV. CLUSTERING RESEARCH PAPERS BASED ON TOPICS

Till now, we have *closed frequent keyword-set*s, each representing a cluster of research papers. These topic clusters are complete in the sense that we have the maximal length *keyword-set* shared by all the research papers represented by that topic. In the mapping *Keyword Set Paper List*, we have the list of papers corresponding to a *topic*. These sets of papers in the list form different clusters. These clusters of research papers are overlapping in nature as a paper may span more than one topic. These clusters are also hierarchical in nature. The cluster representing a broader topic is essentially a combination of several clusters representing its sub-topics, which in turn are a combination of the sub-sub-topic clusters and so on. For example, *databas system* is a broad topic and *imag databas systems, distributed databas systems,* etc. are sub-topics lying within the broader topic *databas system.* Each level of the hierarchy represents a different level of data description, which facilitates the knowledge discovery at various levels of abstraction.

Thus, we have used *closed frequent keyword-sets* to form topics and used these topics as the similarity measure to cluster the research papers.

V. RANKING OF RESEARCH PAPERS

For each topic we have a cluster of research papers in which the topic lies. To find out which research papers are of good quality, we have developed a time independent, modified iterative *PageRank* algorithm. We ranked the research papers based on citation network. Each research paper is cited by a number of research papers and there exists a well-defined graph structure among the network of research papers.

The basic algorithm for ranking the research papers based on citation network uses the two types of edges in a graph: Outlinks and Inlinks.

Definition 5. Outlinks: *From a given node N, link all the nodes N_i that the node N cites.*

Definition 6. Inlinks: *To a given node N, link all the nodes N*j *that cite the node N.*

These *outlinks* and *inlinks* will be used while calculating the authoritative score [12] for each node. The procedure 2 uses modified iterative PageRank algorithm to calculate the authoritative score for each node.

Procedure 2: Time-independent Modified Iterative PageRank Algorithm
Require: Citation Network *CN*, Paper Year *PY*, Outlinks Count *OC*, Paper Inlinks *PI*, Damping Factor θ

```
 1: for each paper R in PY
 2:     year Y = PY[R]
 3:     Year Citation Count YCC[Y] += OC[R]
 4:     Year Paper Count YPC[Y] += 1

 5: for each year Y in YCC
 6:     Average Year Citations Count AYCC[Y] = YCC[Y]/YPC[Y]

 7: Initialize Paper Rank PR to 1.0 for each paper R

 8: while true
 9:     flag = true

10:     for each paper R in PR
11:         Current Score CS = PR[R]
12:         if R in PI
13:             Inlinks List IL = PI[R]
14:             New Score NS = 0.0
15:             for each inlink I in IL
16:                 if I in PR
17:                     NS += PR[I]/OC[I]
18:             year Y = PY[R]
19:             NS = (1-θ) + θ*NS/AYCC[Y]
20:             if CS is not equal to NS
21:                 flag = false
22:             Updated Paper Rank UPR[R] = NS

23:     if flag is equal to true
24:         break
25:     copy UPR to PR
26:     clear UPR
27: Maximum Score MS = Maximum Score in PR
28: for each paper R in PR
29:     PR[R] /= MS
```

In the procedure *2*, we have the citation network *CN* as input from which we create an Outlink Count *OC* for each research paper which keeps track of the number of *outlinks* corresponding to each research paper. We also create Paper Inlinks *PI* which maps each paper to the list of papers which are its *inlinks*. In the PageRank algorithm, the damping factor θ is also used which is required to prevent the scores of research papers that do not have any *inlinks* from falling to zero. For the experiments we set the damping factor to *0.85* [5] which gave satisfactory result. In steps *1-6,* we calculate the value of metric Average Year Citations Count *AYCC* which is the metric we introduce to counter the time dependence of PageRank algorithm. This metric is *the average number of citations per paper in a particular year* which is a time dependent metric and directly reflects the varying distribution of citations over the years. We observe that this metric captures the time bias against the newer papers well and has high values for older papers and low values for newer ones. Considering the year of publication of all the research papers, we pre-compute the total number of citations for each year and the number of research papers published in each year. Using them, the average number of citations per paper for each year is determined.

In line *7,* each paper's score is initialized to *unity* and then the algorithm iteratively modifies the score depending on the score of other papers that point towards it. It stops when all the research paper's scores converge, i.e. become constant**.** *Step 8* to *step 26* is the iterative calculation of the authoritative score for each research paper *R*. The iteration stops when there is no change in the score of any *R*. This is signified by no change in the value of *flag,* set to *true* in *step 9; flag* is set to *false* in *step 16* if there is a change in the score of any paper *R* during the update step. From *step 10* to *step 22*, for each research paper, a new authoritative score is calculated based on the scores of the *inlinks* in the previous iteration. The PageRank algorithm is based on the fact that the quality of a node is equivalent to the summation of the qualities of the nodes that point to it. Here, quality refers to the score of the research paper. This fact is used in *step 17* by dividing the *inlink* score by *OC*[*I*] which is the number of *outlinks* of the *inlink*. This takes care of the fact that if a research paper cites more than one paper, it depicts that it has drawn inspiration from various sources and hence its effect on the score of the paper it cites should diminish by a factor equal to the number of paper it cites. *Step 19* modifies the score calculated above to incorporate the damping factor θ. It also incorporates the time-independence factor by using the *average citations per paper in a year* metric *AYCC*[*Y*]. In *step 23-24*, if the scores of none of the papers differ from the previous iteration, the value of *flag* is unchanged from its *true* value, and the loop breaks indicating the convergence of the algorithms. In *step 27* to *29*, we normalize the scores to scale down the scores within the range *[0, 1]*. Thus, finally we get the authoritative score of each research paper *R* based on the score of the research papers citing it.

VI. EXPERIMENTS AND RESULTS

A. Dataset

To show the results of our algorithms, we used the *DBLP XML Records* available at *http://dblp.uni-trier.de/xml/* [7]. The *DBLP* dataset contains information about various research papers from various fields published over the years. This information includes the title of the research paper, its author(s), the year of publication, the conference of publication, a unique key for each research paper and the keys of the research papers the given research paper cites. It is to be noted that the dataset used by us contained research papers with citation information till the year *2010* only.

B. Data Preprocessing

The *DBLP* dataset also contains information that is not useful in our algorithms. We need to pre-process the dataset to extract only the information that we will use in our algorithms. In data pre-processing, we extracted all the research paper titles, year of publication and citations. Also, all the keywords present in the titles of the research papers were stemmed using the Porter's Stemming Algorithm.

C. Results

We implemented our algorithms on the *DBLP* dataset and discovered various interesting results. It contains *16,32,442* research papers from various fields of research. An objective and quantitative evaluation of the result is difficult due to the lack of standard formal measures for topic detection tasks. But, the list of topics generated by our experiments on examination by field experts and based on our observations match prevailing topics in the dataset.

D. Topic Modeling

The topic modeling algorithm consisted of forming *closed frequent keyword-sets* extracted from the phrases present in titles of the research papers in the *DBLP* dataset. Each of the *closed frequent keyword-sets* represents a distinct *topic*.

We tested our algorithms on various values of minimum support. Upon implementing the algorithms with minimum support *100*, we obtained *12,057* topic clusters consisting of *5,476* 1-length topics, *5,766* 2-length topics, *748* 3-length topics, 62 4-length topics and *5* 5-length topics. Based on their support, the top three *n*-length topics were:

TABLE I: TOP 3 n-LENGTH TOPICS BASED ON SUPPORT

n-length	***Top Three Topics***		
1-length	*system*	*model*	*network*
2-length	*neural network*	*real time*	*sensor network*
3-length	*wireless sensor network*	*support vector machin*	*ad hoc network*
4-length	*mobil ad hoc network*	*wireless ad hoc network*	*content base imag retriev*
5-length	*radial basi function neural network*	*ieee 802 11 wirelss lan*	*low density pariti check code*

D.1. Top Research Papers in a Topic Cluster

For each topic *T*, we have the ranked list of research papers as determined by our algorithms. This can be very useful to researchers studying a given topic *T*.

Following is the list of the top three research papers for some of the prominent topics of research.

TABLE II: TOP 3 RESEARCH PAPERS FOR SOME PROMINENT TOPICS ALONG WITH THE TOPIC CLUSTER SIZE

Topic	***Cluster Size***	***Top 3 Research Papers***
algorithm	*81506*	*Introduction to Algorithms.*
		Graph-Based Algorithms for Boolean Function Manipulation.
		Fast Algorithms for Mining Association Rules in Large Databases.
associ rul	*1737*	*Mining Association Rules between Sets of Items in Large Databases.*
		Fast Algorithms for Mining Association Rules in Large Databases.
		Fast Discovery of Association Rules.
onlin social network	*104*	*A familiar face(book): profile elements as signals in an online social network.*
		Information revelation and privacy in online social networks.
		Measurement and analysis of online social networks.
neural network	*19178*	*Neural Networks and the Bias/Variance Dilemma.*
		Image processing with neural networks - a review.
		Evolving Neural Network through Augmenting Topologies.
xml databas	*226*	*TIMBER: A native XML database.*
		Efficient Keyword Search for Smallest LCAs in XML Databases.
		Querying Structured Text in an XML Database.

The top research papers for the topics were found to match the most *popular* research papers from that topic as determined by field experts. Also our results were found to be considerably consistent irrespective of the cluster size or the age of the topics.

E. *Performance*

To evaluate the performance of our algorithm, we varied *minimum support* and plotted the time taken by our algorithm to run. The algorithm builds the *frequent keyword-sets* and eliminates the non-closed frequent ones. The following graph shows this variation.

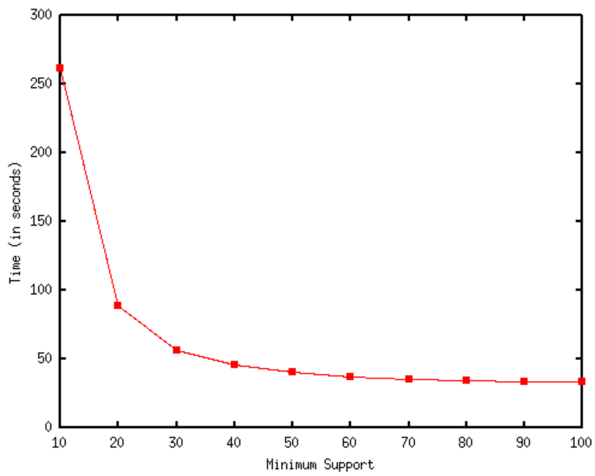

Figure 2. Graph showing minimum support versus time taken by the algorithm to run

Above algorithm was run on a *3GB* machine with *2.10GHz Intel® Core™ 2 Duo CPU.*

VII. APPLICATIONS

Our algorithms have a variety of applications in various fields of research. Following are some of the applications areas:

- *Topic Search System:* This system would retrieve the top ranked papers of a given topic. The retrieval of papers on year-wise granularity could be given as an option.
- *Topic Ranking System:* Using the cluster of research papers for the topics and their authoritative scores, we would to like build a system that would compare the topics and produce a ranked list of topics.
- *Evolution of Topics:* Evolution of topics is a useful tool for new researchers. He can intuitively deduce the emerging fields of research by seeing the trends in the currently *popular* topics.
- *Recommendation Systems:* Extending the clustering to authors and conferences based on topics, we can build a recommendation system. By comparing the topics of interests of the authors, we can recommend those topics to an author on which other *similar* authors have worked. This *similarity* between authors could be based on a threshold number of common topics of research shared by them. Similarly, we can also recommend the conferences of interest to the authors.
- *Finding Landmark Papers:* Since a topic cluster contains all the research papers published on the topic, we can determine the most *important* papers for the topic by observing the graph of evolution of the topic. A sudden peak in the graph would suggest the presence of an *important* paper for the topic. These *important* papers could be recognized by comparing the authoritative scores of the papers in the corresponding topic cluster and then be classified as *Landmark Papers*.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a method to derive topics and cluster research papers into these topics. The topics were identified by forming *closed frequent keyword-sets* as proposed by our algorithms, which works better than traditional approaches like *Apriori.* We also proposed a method to produce the ranked list of research papers within a topic cluster. We analyzed the results of topic modeling as well as the performance of our algorithms on varying the minimum support parameter.

As mentioned above, our algorithms have a variety of applications. In future, we would like to implement our algorithms to build systems for topic search, topic ranking and recommendations. We would also like to examine statistical approaches for topic correlation and explore other domains like web site clustering, document clustering, etc. in which our algorithms can be applied.

IX. REFERENCES

[1] R. Agarwal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proceedings of the 1993 ACM SIGMOD Conference, 1993.

[2] R. Agarwal, and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proceedings of the 20th VLDB Conference, 1994.

[3] F. Beil, M. Ester, and X. Xu, “Frequent Term-Based Text Clustering,” Proceeding of the 8th International Conference on Knowledge Discovery and Data Mining (KDD), 2002.

[4] D. M. Blei, and J. D. Lafferty, “Correlated Topic Models,” NIPS, 2005.

[5] S. Brin, and L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” Proceedings of the 7th International Conference on World Wide Web, 1998.

[6] Z. Chong, L. Yansheng, Z. Lei, and H. Rong, “FICW: Frequent Itemset Based Text Clustering with Window Constraint,” Wuhan Journal of Natural Sciences, Vol: 11, No: 5, pp: 1345-1351, 2006.

[7] The DBLP Computer Science Bibliography. http://dblp.uni-trier.de/

[8] E. Erosheva, S. Fienberg, and J. Lafferty, “Mixed-membership Models of Scientific Publications,” Proceedings of the National Academy of Sciences, 2004.
[9] X. Geng, and J. Wang, “Toward theme development analysis with topic clustering,” Proceedings of the 1st International Conference on Advanced Computer Theory and Engineering, 2008.
[10] T. I. Griffiths, and M. Steyvers, “Finding Scientific Topics,” Proceedings of the National Academy of Sciences, 2004.
[11] Y. Jo, C. Lagoze, and C. L. Giles, ”Detecting Research Topics via the Correlation between the Graphs and Texts,” Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007.
[12] J. Klienberg, “Authoritative sources in a hyperlinked environment,” Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998.
[13] J. Kleinberg, “Bursty and Hierarchical Structure in Streams,” Proceedings of SIGKDD, 2002.
[14] S. M. Krishna, and S. D. Bhavani, “An Efficient Approach for Text Clustering Based on Frequent Itemsets,” European Journal of Scientific Research, 2010.
[15] C.Y. Lin, and E. Hovy, “The Automated Acquisition of Topic Signatures for Text Summarization,” Proceedings of the COLING Conference, 2002.
[16] E. Lloret, “Topic Detection and Segmentation in Automatic Text Summarization,” http://www.dlsi.ua.es/~elloret/publications/SumTopics.pdf, 2009.
[17] G. S. Mann, D. Mimmo, and A. McCallum, “Bibliometric Impact Measures Leveraging Topic Analysis,” JCDL, 2006.
[18] A. McCallum, A. Corrada-Emmanuel, and X. Wang, “The Author-recipient-topic Model for Topic and Role Discovery in Social Networks: Experiments with Enron and Academic E-mail,” Technical Report, 2004.
[19] Q. Mei, and C. Zhai, “Discovery Evolutionary Theme Patterns from Text – An Exploration of Temporal Text Mining,” Proceedings of SIGKDD, 2005.
[20] N. Pasquier, Y. Bastide, R. Taoull, and L. Lakhal, “Efficient Mining of Association Rules Using Closed Itemset Lattices,” Information Systems, 1999.
[21] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. I. Griffiths, “Probabilistic Author-topic Models for Information Discovery,” Proceedings of SIGKDD, 2004.
[22] X. Wang, and A. McCallum, “Topics over Time: A Non-Markov Continuous-time Model for Topical Trends,” Proceedings of SIGKDD, 2006.
[23] C. Wartena, and R. Brussee, “Topic Detection by Clustering Keywords,” Proceedings of the 19th International Conference on Database and Expert Systems Applications, 2008.
[24] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha, “Probabilistic Models for Discovering E-communities,” Proceedings of WWW, 2006.
[25] L. Zhuang, and H. Dai, “A Maximal Frequent Itemset Approach for Web Document Clustering,” Proceedings of the 4th International Conference on Computer and Information Technology, 2004.