ISSN:0975-9646

Efficient Topic Evolution Discovery in a Corpus using Semantic Similarity and Citation Graph

Vijay Subramanya[#1] Saiprashanth Kote[#2] Santosh Kumar[#3] P. Santhi Thilagam[#4]

#Department of Computer Science and Engineering National Institute of Technology Karnataka, Surathkal Srinivasnagar, Mangalore, India - 575025

***Abstract*—A topic evolution graph enables one to get a quick overview of the knowledge body and its growth. Hence, discovery of topic evolution over time in a scientific corpus is an important problem. We propose a technique that uses the content of documents to discover topics as well as to link them in order to represent evolution. In addition to depicting the growth of knowledge body as a topic evolution chart, we mine influential documents and extract representative documents for topics. Previous work includes citation-aware approaches that go beyond considering documents as bags-of-words but which are computationally expensive. We discover topic labels (n-grams) using semantic relevance measures and discover influential nodes by applying the PageRank algorithm on the citation network. We evaluate our method on an arXiv corpus of ~29,000 research papers in physics. While producing meaningful results, our technique is also found to offer improvement on existing methods in terms of computational complexity and scalability. Specifically, our method runs in time linear in the number of documents and words in the corpus.**

***Keywords*—Topic evolution, corpus, citation graph, influential documents.**

I. Introduction

One of the major obstacles a person new to a body of knowledge faces is the lack of a bird's eye view of the area over a duration. Such an overview would help her get a sense of how the area has evolved from a basic, foundational idea and how the recent developments are connected to earlier work. For instance, in the case of a researcher looking to enter a new field of research, an evolution tree of the most influential publications saves much time and effort in visualizing the growth of the topic and identifying the papers to read. In addition, the leaves of the tree indicate the current research topics in the area and help in choosing the topic of research.

Topic evolution can be applied in domains of knowledge other than scientific literature, provided the citation information is available. On the web, links in the web pages act as citations to other pages. This can be used to build a citation network and subsequently discover the evolution of a topic in an online corpus (ex: wikipedia). In particular, this method would be of great value in analyzing the growth of news stories by linking a current news event to past events that could have influenced it.

The first step is the discovery of topics in the corpus. Latent Dirichlet Allocation has been used for this purpose in the past [3], [5]. Recently, citation information has also been used to build topic models [5], [6], while at the same time dispensing with time-based pre-division of the corpus [6]. However, these methods tend to be computationally expensive and unsuitable for online systems where efficiency is a major factor. We adopt the concept of word association [15] to eliminate meaningless terms and score the remaining terms for semantic relevance to the document collection [16]. The terms with highest scores are selected as topic labels for that document collection. Before applying this method on the corpus, we group the documents in each year based on their content (title and abstract). The L-method [12] gives us the optimum number of clusters which is then input to the hierarchical clustering algorithm.

Once the topic labels are obtained for each cluster, the next step is linking those clusters to depict evolution. We describe our way to measure significance of cross citations between clusters and use these significance values along with topic similarity scores to determine the parent of a given cluster in the evolution graph. Further, representative documents in each cluster can be mined which help depict the evolution better and adds to the information presented to the user. For this task, we use the Google PageRank algorithm to measure influences of papers based on the weight of citations a document has received.

We have implemented our method on MATLAB and run it on an archive of physics papers published in the arXiv journal during 1992-2003. The results are impressive in terms of correctness and sufficient detail. The topics detected are found to be relevant to the research and evolution links effectively portray the growth of topics. Moreover, our method is shown

to run in time linear in the number of documents and terms in the corpus.

II. Literature Survey

Topic modeling is divided into two branches: topic detection and topic evolution. Our work uses concepts of both the areas to generate topic evolution graphs. Furthermore, we mine influential documents as an additional detail to provide to the user.

A. Topic Detection

Topic detection involves discovering topics that effectively represent a set of documents and has been studied extensively [1]. In one of the pioneering works in topic detection, Lin and Hovey [2] presented a method for automatic extraction of topic signatures from text which could be used for summarization. Their method was found to be better than *tfidf* and baseline algorithms. Griffiths and Steyvers [3] used the Latent Dirichlet Allocation model to select topics based on the similarity between the word distribution of the corpus and the the word distributions the candidate topics generate.

The above methods do not utilize the citation network to discover topics. Jo et. al. [4] include the citation information and score a bigram based on the connectivity of the subgraph in the citation graph induced by the documents containing the bigram. The relevance of the term is determined by comparing the density of the subgraph induced by documents containing the term and a randomly chosen subgraph. He et. al. [5] further incorporated citations in their generative topic model. To discover topics in a time period, in addition to documents in that period, they also consider the documents cited by those documents.

Previous work also differs in whether the documents are time stamped continuously or are segregated into discrete time blocks. While He et. al. [5] assume pre-divided time units in the corpus, Jo et. al. [6] make no such assumptions. They model the entire document collection and discover topics without imposing homogenetic or topological restrictions on the corpus.

However, finding that analyzing the entire corpus as a whole could be computationally inefficient, we choose to adopt the pre-divided corpus approach of He et. al [5]. Our results show that there is sufficient variation in the topic labels when time unit is taken to be one year.

B. Topic Evolution

The discriminative approach to topic evolution treats each topic as a word distribution over a collection of documents belonging to a time unit. Morinaga and Yamanishi [7] use finite mixture models to represent documents. A significant change in the topic mixtures indicates evolution of the topic. Mei and Zhai [8] conduct clustering sequentially and then correlate clusters via a temporal graph model. Schult and Spiliopoulou [9] use a clustering approach to discover the ontology/taxonomy evolution for documents.

Recently, many studies used generative topic models to observe topic evolution on document streams. Zhou et al. [10] used the LDA model to observe temporal topic evolution over scientific literature. Specifically, a *k*-component LDA model is constructed over the whole dataset to generate *k* global topics. For each topic, the trend is obtained by simply counting the number of papers be- longing to the topic year by year. Mann et. al. [11] used an n-gram topic model to identify the influence of one topic on another. However, this approach modelled citations indirectly in the topic model, and the resulting topic influence is also time irrelevant. Recently, He et. al. [5] consider both content and citations in their inheritance topic model. Moreover, they use a citation network analysis approach to explicitly emphasize the relationship between topics. Jo et. al. [6] derive a measure that uses cross citation count between the documents of two topics to link them.

In our work, looking to minimize computational complexity, we use the cross citation count, influence of citing and cited documents and topic similarity to determine “evolved-from” relationship between document clusters (topics). This approach is validated by our results.

C. Mining Representative Documents

If each node in the topic evolution graph is associated with, in addition to topic labels, a few representative documents, the user can quickly browse through these documents without having to search for documents in each cluster (node). To achieve this, we borrow ideas from influential node mining, a well-researched problem in the graph mining area.

Katz [18] first proposed a method to measure the standing of a node in a social network using information about the number of paths and their lengths terminating at that node. The standing of a node is the sum of contributions from all other nodes where the individual contribution decreases exponentially with increase in path-length. Hubell [24] later included link-weights to improve the estimate. The standings of nodes in his method were solutions to a system of equations that related a node’s standing to the standing of its neighbours.

In bibliometrics, Garfield’s impact factor[19] is a popular measure of a journal’s influence. However, it focuses only on the number of citations a journal has managed to obtain in the prevous two years. Moreover, the influence of the citing journals is not considered.

Kleinberg [21], borrowing ideas from Katz and Garfield, extended the work on academic networks to hyperlinked environment of the World Wide Web. He proposed algorithmic methods to discover authoritative information sources on broad search topics on the Web. Brin and Page [20], in their seminal paper that led to the birth of Google, used the above concepts and presented the PageRank algorithm to objectively measure the global “importance” ranking of a webpage using the link structure of the Web. More recently, Chen et. al [17]. showed

that the PageRank algorithm, when applied on citation graphs, is capable of mining the most influential research papers, even when they are not heavily cited. This is because the PageRank algorithm considers both the influence of the citing paper and the number of papers the citing paper has cited which leads to better estimates of impact of a paper in its field.

Following Chen et. al. [17], we use the PageRank algorithm to mine influential documents from the corpus.

III. Problem Definition

A. Topic Detection

Given a corpus and a candidate topic label θ, the topic detection problem is to determine if the word distribution generated by θ is similar to that of the entire corpus beyond a threshold.

B. Topic Evolution

Following He et. al. [5], we define the topic evolution problem as follows.

The corpus D is divided temporally into exclusive subsets D_1, D_2, ..., D_n. Let θ_t be the topics discovered in D_t. Given a topic z at time t, the topic evolution problem is to assign a topic y in θ_{t_1}, $t_1 < t$, as the parent of z such that the probability of topic z having evolved from topic y, $P(z|y)$, is maximized.

C. Mining Influential Documents

Given a collection of documents D and the citation graph G_c, we want to determine the influences of documents. The influence of a document d_i depends on the influences of the documents that cite it. The contribution of a citing document d_j varies inversely as the number of documents cited by d_j.

D. System Description

The input is a collection of documents D that is divided into discrete time units and a citation network. The documents at time t are denoted by D_t. Each document is an ordered set of words. The citation network is a directed graph $G_c(V,E)$ wherein an edge from node u to node v denotes that the document v is cited by document u.

The objective is to discover the evolution of topics in the corpus D and depict it using topic evolution graphs. Also, influential documents are to be mined to better represent the evolution.

The output is a topic evolution graph in which each node is associated with a document cluster which is represented by a list of topics and representative documents.

Assumptions: The documents are time stamped. Therefore, the corpus can be pre-divided into time units. We also assume that the time slots are of equal duration.

IV. Methodology

The steps involved in generating a topic evolution graph with representative documents (output) from a corpus and the corresponding citation graph (input) are depicted in Figure 1.

A. Grouping Documents

The first task is to discover the topics that represent the collection of documents in a time unit t. Since there may be documents belonging to vastly differing topics in the same time unit, we must first group the documents in D_t according to their topics. Before proceeding with clustering, as a preprocessing step, we build the term-document matrices for unigrams, bigrams and trigrams in D_t. These matrices are used for further computation, and henceforth the original documents will no longer be accessed.

1) Singular Value Decomposition: The term-document matrix gives the relationship between words and documents. In order to compare documents, we need to extract document-specific information from the term-document matrix. Singular Value Decomposition (SVD) is a mathematical technique that helps reduce the number of columns (i.e. terms) while retaining similarity structure among rows (documents).

A decomposition of a matrix X into orthogonal matrices U and V and a diagonal matrix Σ is called a singular value decomposition.

$$X = U\Sigma V^T$$

To get a rank k approximation to X, we select the k largest singular values from Σ and the corresponding rows from U and V. In other words, the rows of U and V are approximations of the terms and documents respectively in a lower dimensional space. In order to compare any two documents, p and q in this lower dimensional space, we only have to compare (using cosine similarity) their corresponding rows in V. Once we have defined the similarity measure for two documents, we can apply clustering on the document space to group the documents.

2) Clustering: Unsupervised clustering is used to group documents based on the cosine similarity of their lower dimensional approximations. Since partitioning based clustering algorithms such as k-means do not scale well for large datasets, we adopt hierarchical clustering.

One major concern in unsupervised clustering is deciding the number of clusters to be formed. Knee-refinement is a well-known method for determining the number of clusters. Salvador and Chan [12] have proposed an efficient algorithm for finding the knee in the evaluation curve known as the *L-method.* The evaluation graph plots the distance between the two most similar clusters (y-axis) in a clustering of a given number of clusters (x-axis). The *knee* of the evaluation graph is the point of maximum curvature. The L-method uses the fact that the regions on either side of the knee are approximately linear. The point of intersction of the two best-fit lines is found and the x-value at this point gives the optimum number of clusters to be formed.

The number of clusters found by the knee-refinement method is then input to the hierarchical clustering algorithm along with the documents represented by the matrix V to obtain clusters.

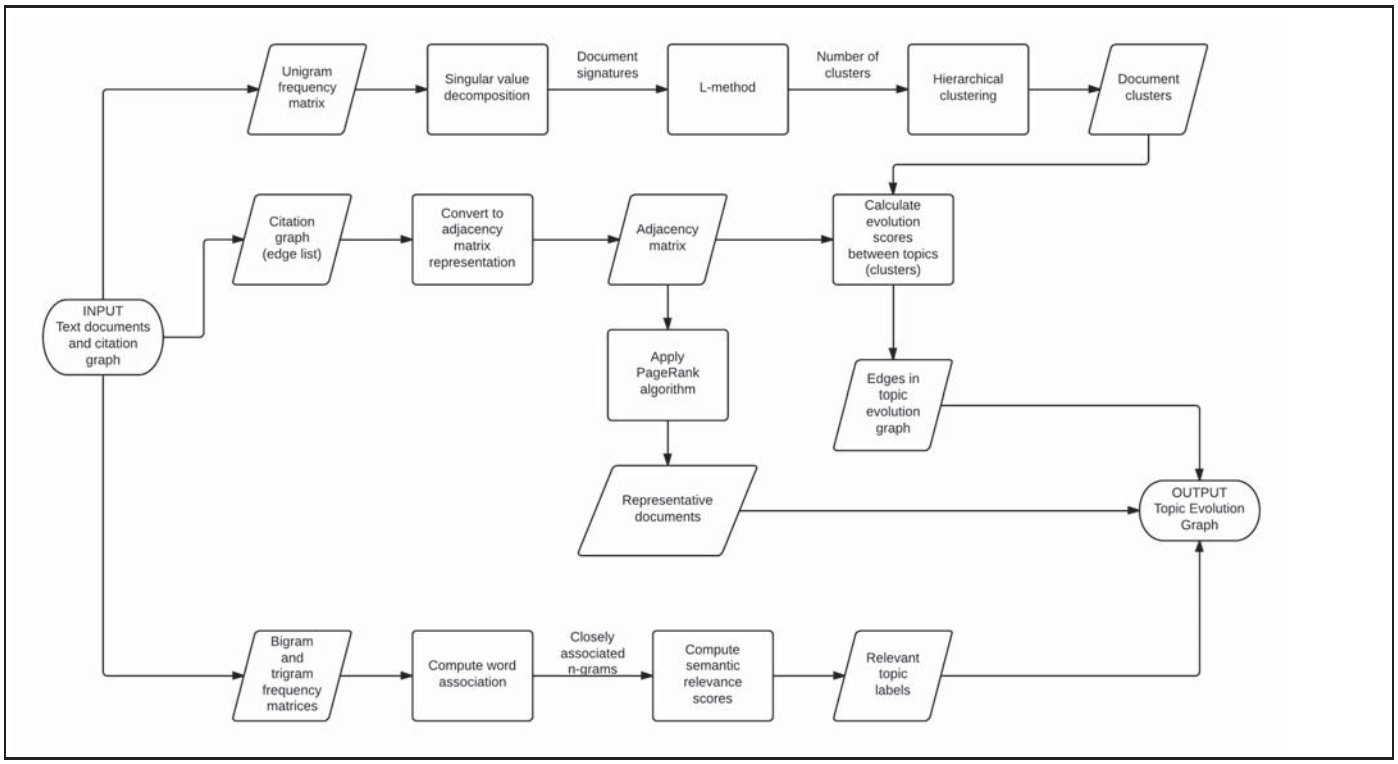


Figure 1: Flow diagram of execution steps

B. Topic Detection

In this subsection, we consider a corpus and ask how to discover topics in this document collection.

Bigrams and trigrams have been shown to be useful in many text mining applications. Seymore and Rosenfield [13] use bigrams and trigrams, along with tf-idf, for large-scale topic detection. Barrón-Cedeño et. al. [14] have developed a method for plagiarism detection using bigrams and trigrams. Thus, it is reasonable to assume that searching for topics in the bigram- and trigram-space produces sufficiently accurate results.

1) Meaningful n-grams: How to eliminate meaningless bigrams and trigrams? The association ratio defined by Church and Hanks [15], based on mutual information, provides a means for achieving it. Association ratio of two words, x and y, is defined as

$$A(x,y) \quad = \quad \log_2 \frac{P(x,y)}{P(x)P(y)}$$

where $P(x)$ and $P(y)$ are the probabilities of finding the words x and y in the corpus; $P(x,y)$ is the probability of finding the bigram xy in the corpus. Association ratio can be extended to the case of trigrams similarly. Notice that the numerator tells us how likely we are to see x and y together in that order and the denominator normalizes this value to prevent common bigrams such as “of course” and “in addition” from having very high association ratios.

Once the association ratios are computed for all n-grams, only the n-grams with high association ratios can be kept for further consideration. This helps us eliminate bigrams and trigrams that are not *closely related*.

2) Selection of Topic Labels: At this point, we have a list of meaningful bigrams and trigrams obtained from the corpus. The task now is to select those n-grams that best represent the corpus.

We use the *Relevance Score* measure defined by Mei et. al. [16] to find the topic label that is most similar to the topic model. The relevance score of a label to a topic model, $s(l,\theta)$, is a measure of the semantic similarity between the label l and the topic model θ. Supposing that l is a meaningful term that has passed the association ratio test, $s(l,\theta)$ tells us how appropriate a label l is for θ. The two relevance scoring functions they define are

1) *Zero-Order Relevance*: The semantic relevance score of a candidate label $l = u_0u_1...u_m$ is defined as

$$Score_0 \quad = \quad \sum_i \log \frac{p(u_i|\theta)}{p(u_i)}$$

In this measure, labels containing more important words in the topic distribution are considered relevant. That is to say, a candidate label is scored based on the likelihood that the individual words in the label generate the topic model. This method does not make use of context information that can be obtained from the corpus. Moreover, it fails to recognize that the semantic information carried by the label as a whole may be significantly different from what is conveyed by its parts. It is possible that a label constructed from high probability words makes

no sense in the context of the topic. This limitation is overcome by the First-Order Relevance.

2) *First-Order Relevance*: Context is essential when interpreting the semantics of a topic model to extract labels. Accurate labels for a topic θ must have high probabilities of occurring in documents that cover θ. One way of achieving this is to compare the multinomial word distributions of candidate labels with the topic model distribution to determine their semantic similarity with the topic. The first-order relevance of a label l to a topic θ is given by

$$Score_1 = -\sum_w p(w|\theta)\log\frac{p(w|\theta)}{p(w|l)}$$

where $p(w|\theta)$ is the probability of the word w in the multinomial distribution of topic θ, and $p(w|l)$ is the probability of w in the multinomial distribution of the label l. However, we have no way of knowing this multinomial distribution. Hence, we approximate it to $p(w|C)$ where C is the set of documents in which the label l appears. The document collection C represents the context of the label l.

Since the first-order relevance score takes into account the relevance of words in the context of a label, it discovers topic labels much more accurately than the zero-order relevance. Therefore, we choose to adopt first-order relevance to find the relevance of the meaningful n-grams obtained in the previous step. The most relevant n-grams are selected as topic labels for the corpus.

C. Topic Evolution

Citation information is used to determine the "evolved-from" links in the topic evolution graph. Basically, nodes in the evolution graph are clusters of documents which are labeled by topic labels discovered as in the previous section. Specifically, for a node z at time t, we want to find the node in the evolution graph from which the topics in node z have most likely evolved. There are two things to note here. First, any link in the evolution graph must be based on the significance of citations between the nodes. Second, if there are two nodes, x and y, which have significant citations from node z, then the node whose topics differ the least from those of node z, say y, is the immediate parent of z. This means, we must introduce a link from z to y and not from z to x.

1) Significance of Citations: The significance of citations from documents in node z to documents in node y depends

1) *inversely on the influence of cited documents in node y.* Greater the influence of cited documents, less likely it is that the documents in z have directly evolved from those of y. Greater influence of a document means a number of other documents might have evolved from it, so the chances of a particular set of documents (those in z) having evolved from it are low.
2) *directly on the influence of citing documents in node z.* The node cited by the more influential nodes in z is chosen over the node cited by less influential nodes.
3) *directly on the fractions of citations given and received by documents in nodes z and y.* Higher the number of citations, lesser is the strength of each citation.

Now, we can define the significance of citations from node z to node y as follows.

$$Sig(z,y) = \sum_{(i,j)\in E(G_c)} \frac{G_i}{G_j}.\frac{1}{c_{out}(i).c_{in}(j)}$$

where $E(G_c)$ is the set of edges in the citation graph G_c, i is a document in node z, j is a document in node y, $c_{out}(i)$ is the number of outgoing citations at document i, $c_{in}(j)$ is the number of incoming citations at document j. The influences of documents, G_i and G_j, will be defined in the next section.

2) Similarity of Topics: Citations alone may not reveal the entire information about topic evolution. Thus, for those nodes that have significant citations from node z, we compare their word distributions with that of z. The most similar node is chosen as z's parent and a link is added from z to that node.

The citation significance of the previous step is used as a filtering criterion by setting a threshold that is chosen appropriately. Standard measures of similarity such as Euclidean distance may be used for comparing topic models.

D. Mining Representative Documents

Topic evolution graph provides us only an overview of how the topic has grown over time. For further details, the user would have to look for documents in those topics herself. To ease this process, we mine the highly influential documents from the corpus and include them as additional representatives of nodes in the topic evolution graph.

One of the important indices of a document's influence is the number of citations it has received. Katz's method [18] computes a weighted sum of the contribution from every path ending at a node to determine its standing. Garfield's impact factor [19] for journals also uses only the citation count without considering the influence of citing journals. Recently, Chen et. al. [17] have shown that Google's PageRank algorithm [20] can be adopted to effectively mine "scientific gems", as they call it, from citation graphs. This is because the PageRank algorithm considers both the influence of the citing paper and the number of papers the citing paper has cited which leads to better estimates of impact of a paper in its field. Therefore, we choose the PageRank algorithm to mine influential documents in our work.

In the PageRank algorithm, nodes are discovered based on the influence of its citing papers and the strength of those citations. Given a directed graph of N nodes $i = 1,2,...,N$, with edges representing citations, the Google number G_i for the i^{th} node is defined by the formula

$$G_i = (1-d)\times\sum_{(j,i)\in E(G_c)}\frac{G_j}{k_j}+\frac{d}{N}$$

where $E(G_c)$ is the set of edges of the citation graph G_c, k_j is the out-degree of node j and d is a parameter that is varied according to the application. The summation is done over all the neighbours (citing documents) of i. Chen et. al. [17] reason that the value of d appropriate for citation networks is 0.5.

After obtaining the influence of each document in the corpus, we use these influences for two things. First, to calculate the citation significance between nodes of the topic evolution graph, document influence values are required, as described in the previous section. Second, the most influential documents in a cluster are selected as its representatives. This enables the user to view not only the evolution of topics as depicted in the topic evolution graph, but also the beacons that led the growth of these topics.

Selecting a document as a representative is again based on two factors: influence of the document and the similarity between the topic and the document. Once we compute the influences of every document, the topmost documents in each cluster are chosen as candidates. Among these, those documents that are sufficiently similar to the cluster, as inferred by the similarity of their word distributions, are selected as representatives of that cluster.

V. Implementation Details

A. Dataset

We tested our topic evolution model on a set of scientific publications archived for the KDD Cup 2003 competition [22]. The dataset contains research papers in high energy physics published in arXiv journals during the years 1992-2003. Even though full text of the papers are available, we have implemented our method by considering only the titles and abstracts of the papers as their content. The dataset contains a total of 29,569 papers, and their distribution over year of publication is shown in Figure 2(a). The average number of citations received by documents in each year is plotted in Figure 2(b). There are a total of 352,807 links for an average degree of 12. Like much of the previous work, we used a year as the time unit in our analysis.

B. Environment

The implementation was run on an Intel dual core processor (i3-380M, 2.53GHz, 3MB RAM). Initially, we used Python's scikit-learn toolbox [23] to build the term-document matrices for unigrams, bigrams and trigrams. The rest of the computation and processing was performed on MATLAB 7.12 (R2011a).

C. Input and Output

The input to our method are the text files, each containing title and abstract of a document, and the directed citation graph represented in the edge-list format. The text files are segregated beforehand according to the year of publication.

The output are the number of clusters in each year, the topic labels for each cluster, and the evolution score values for pairs of clusters in different years. The clusters are the nodes in the topic evolution graph and the evolution score is used to form links between clusters.

The term-document matrix of unigrams for a year obtained from Python's scikit-learn toolbox is singular value decomposed into U, Σ, and V, where V is a lower dimensional approximation of the documents in the corpus. V is then fed to L-method function which plots the evaluation graph and computes the knee of the graph using knee-refinement algorithm. Now, having got the number of clusters for that particular year, we apply hierarchical clustering algorithm, specifying the number of clusters to be formed. The output are clusters that represent nodes in the topic evolution graph.

The bigram and trigram document frequency matrices are used to calculate the association ratio for each bi- and trigram. This is used to eliminate meaningless n-grams. The meaningful n-grams are then scored for relevance to the topic (cluster) using first-order relevance measure. The most relevant n-grams are chosen as topic labels (node labels).

The edge-list representation of citation graph is converted to adjacency matrix representation and the PageRank algorithm is applied on it. This gives us the influence of each document in the corpus. These values are then used to discover representative documents of each cluster and, along with topic models, are used to calculate evolution scores for pairs of clusters.

The evolution scores determine links in the topic evolution graph. The clusters, topic labels and the links along with representative documents form the final output.

VI. Results and Evaluation

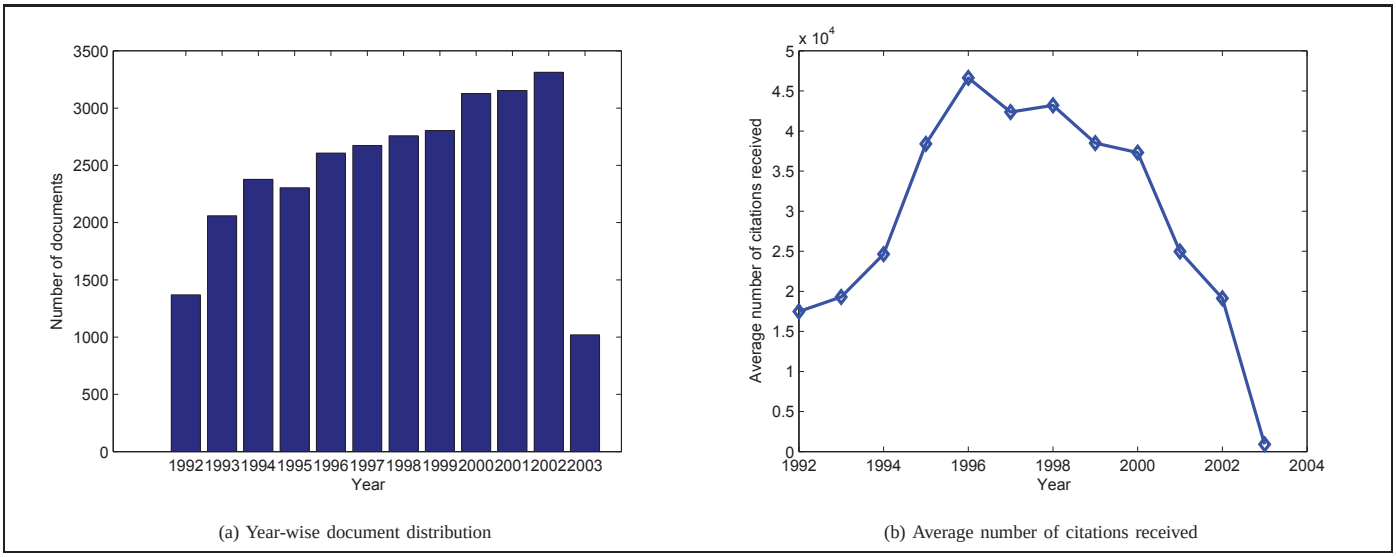
A. Complexity

While analyzing the complexity of our method, it is important to note that we access the documents only once: when building term-document matrices initially. This greatly reduces the complexity.

Time complexity of building term-document matrix for unigrams is, for all practical purposes, almost linear in the total number of words across documents [25]. Because each word of an n-gram appears in at most n terms, building term-document matrices for n-grams can be assumed to be of complexity linear in the number of words too.

Computation of association ratio takes a constant time for a particular n-gram since we only need to look up the term-document matrices to calculate probability of occurrence. Thus, computing association for n-grams takes time linear in the number of n-grams.

To the next step, i.e. scoring the candidate labels for relevance to the cluster, only the meaningful n-grams are input. Generally a threshold on the association ratio is set to exclude meaningless terms. We observed that the association ratio was considerable only for a small fraction (less than 1%) of the terms. Since we analyze only the title and abstract of documents which are usually within a few hundred words, it is reasonable to assume that the number of meaningful terms to be analyzed is of the order of the number of documents



(a) Year-wise document distribution

(b) Average number of citations received

Figure 2: Dataset information

in the collection. Now, for each meaningful term, we add the rows of term-document matrix that correspond to documents containing the term and compare the result with the word distribution of the cluster. When performed in parallel, this step takes $O(d)$ time.

Zhang and Zhu [26] show that approximate SVD of an $m \times n$ matrix can be performed in linear time, i.e. $O(m+n)$. In our case, m is the number of documents and n is the number of terms in the corpus. Proceeding further, hierarchical clustering takes $O(d^2)$ time, where d is the number of documents. However, Zamir and Etzioni [27] have proposed a linear time clustering algorithm based on suffix trees, which can be used to bring the complexity of the clustering step down to $O(d)$.

Since each document cites more or less a constant number of other documents, number of edges in the citation graph can be assumed to be of the order of the number of documents. This means, conversion of edge-list to adjacency matrix can be done in time $O(d)$ and the PageRank runs in $O(d)$ time.

Empirical analysis of time complexity performed on the aforementioned dataset validates our claims. Figure 3(a) measures the execution time for document clustering and label extraction for corpora of various sizes. We observe that it is an almost-linear curve, with a trace of non-linearity introduced probably by the hierarchical clustering step - a bottleneck in our approach. However, the execution time is found to vary linearly with the number of words in the corpus, as shown by Figure 3(b).

Thus, assuming the adoption of linear time clustering methods, the overall complexity of our technique is $O(w+d)$.

B. Scalability

We consider scalability of our method in terms of addition of documents of another year to the existing corpus.

Since clustering is done for each year separately, as we have shown above, the clustering step takes time linear in the number of documents added.

Further, extracting meaningful and semantically relevant topic labels is also performed for each cluster. As previously shown, this step also runs in linear time.

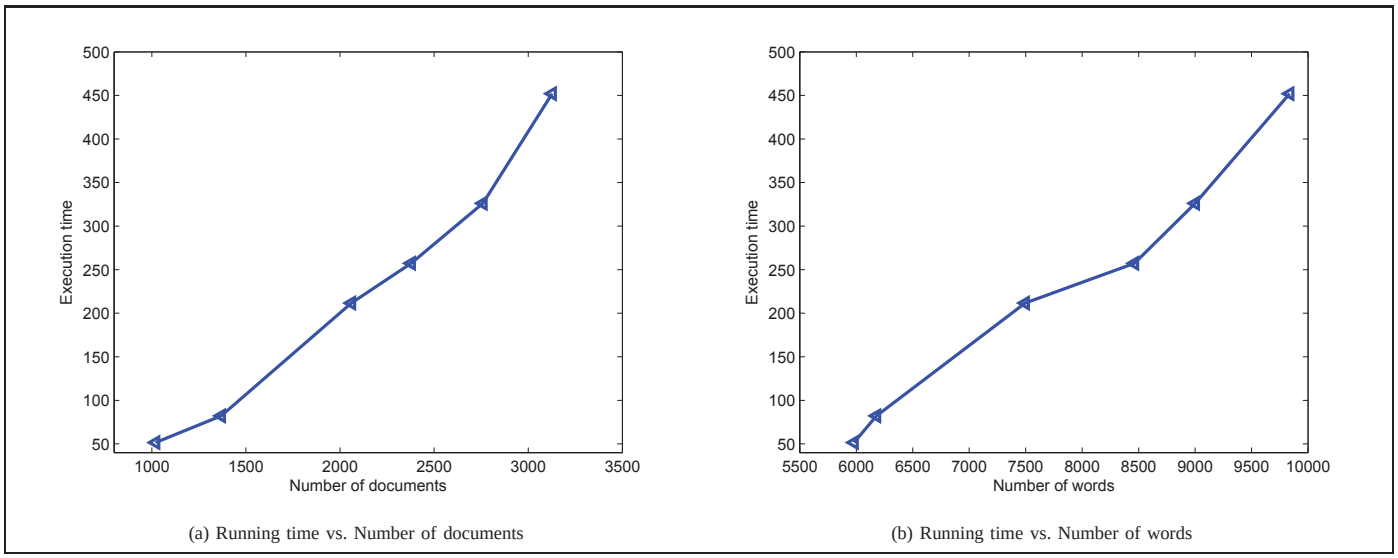
Adjacency matrix can be updated to include new documents without having to change entries in the original matrix. Coming to PageRank, notice that a node's influence can change only when the influences of its citing documents change. Since the new documents added have no citing documents in the citation network, their influences can be approximated to d/N. This addition causes updating of influences of only the nodes cited by those new nodes, which can be done in linear time.

Thus, our method is efficiently scalable (linear time).

C. Case Study

We extracted the titles and abstracts of the arXiv journal papers in high energy physics during 1992-2003 and ran the MATLAB implementation of our method on this corpus. Using the L-method for determining the optimum number of clusters and hierarchical clustering, we obtained clusters for each year ranging from five to nine in number. The top ten topic labels when ordered by semantic relevance scores were chosen to represent each cluster. For a total of 69 clusters across 12 years, 690 topic labels were extracted of which 131 were distinct. Excluding those of 1992, 93 new, distinct labels were extracted over a period of 11 years (an average of 8.45 per year) which is comparable to the average of 9.33 obtained in the citation-aware-Inheritence Topic Model (*c-ITM*) of He et. al. [5] and much better than that of Latent Dirichlet Allocation method (5.93). Further, the number of duplicate topic labels was found to be less than 3% (19 out of 690). More impressive was the fact that only 1.1% (8 out of 690) of labels were noisy or irrelevant.

Average topic similarity between two consecutive years is a good measure of how significantly the topics in a given year differs from those in the previous year. For best results,



(a) Running time vs. Number of documents

(b) Running time vs. Number of words

Figure 3: Time efficiency analysis

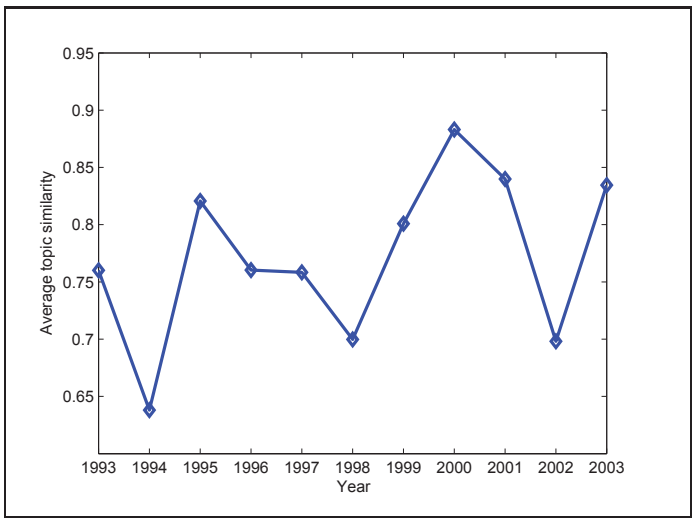


Figure 4: Average topic similarity

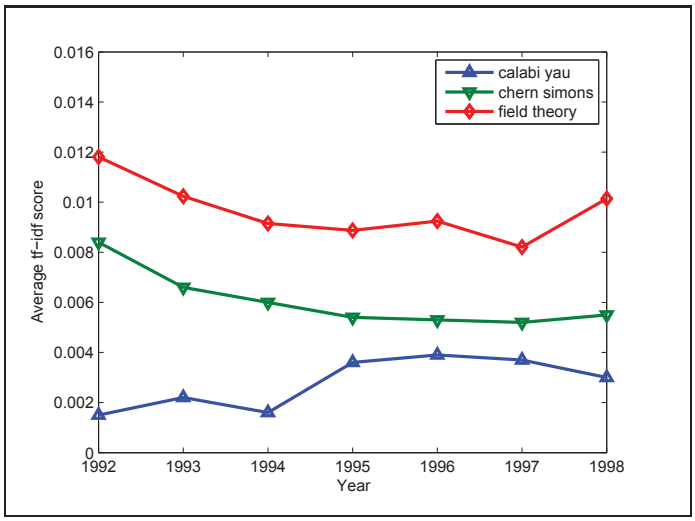


Figure 5: Topic strength trends of *Calibi-Yau*, *Chern-Simons* and *field theory*

a balance needs to be struck between generating new topics and retaining older ones. Following He et. al. [5], we compute the average topic similarity of a year t as follows: for every document in t, find the document in its parent cluster (in $t-1$) that is most similar to it and consider this similarity value. Average topic similarity is the average of these individual similarities. Figure 4 shows the variation of average topic similarity with year. The topic similarity values are observed to be slightly higher than those produced by *c-ITM* of He et. al. [5]. This means, topics tend to retain more in our method when compared to *c-ITM*.

Figure 6 shows a part of the output for the dataset we employed. For easier comprehension, only the portion of the final topic evolution graph that is associated with *field theory* and during the years 1992-1998 is depicted. Clusters are represented by boxes along with their topic labels and arrows are used to represent evolution links. With a limit of 10 labels per box, duplicate labels are removed and the first occurrences of topic labels in a path are italicized.

Since the subject area is *field theory*, the term appears in every box except one in the graph, as expected. Other persistent labels include *gauge theory* which is a type of field theory, *Yang-Mills theory* - a gauge theory that forms the basis of the Standard Model of particle physics, and *string theory* which combines quantum field theory and general relativity. It is interesting to note that the occurrences of *string theory* coincide with the second superstring revolution (1994-2000). Topic evolution is nicely illustrated by the evolution path from *Calabi-Yau* manifold to *anti de-Sitter* space. *Calabi-Yau* was a hot topic in 1996 thanks to Witten's work on Calabi-Yau compactification. This led to work on black holes associated with Calabi-Yau spaces in 1997, most notably the publications of Shmakova and Maldacena et. al. In 1998, Strominger presented an influential paper dealing with black holes whose near-horizon geometries are 3-dimensional *anti de-Sitter* spaces.

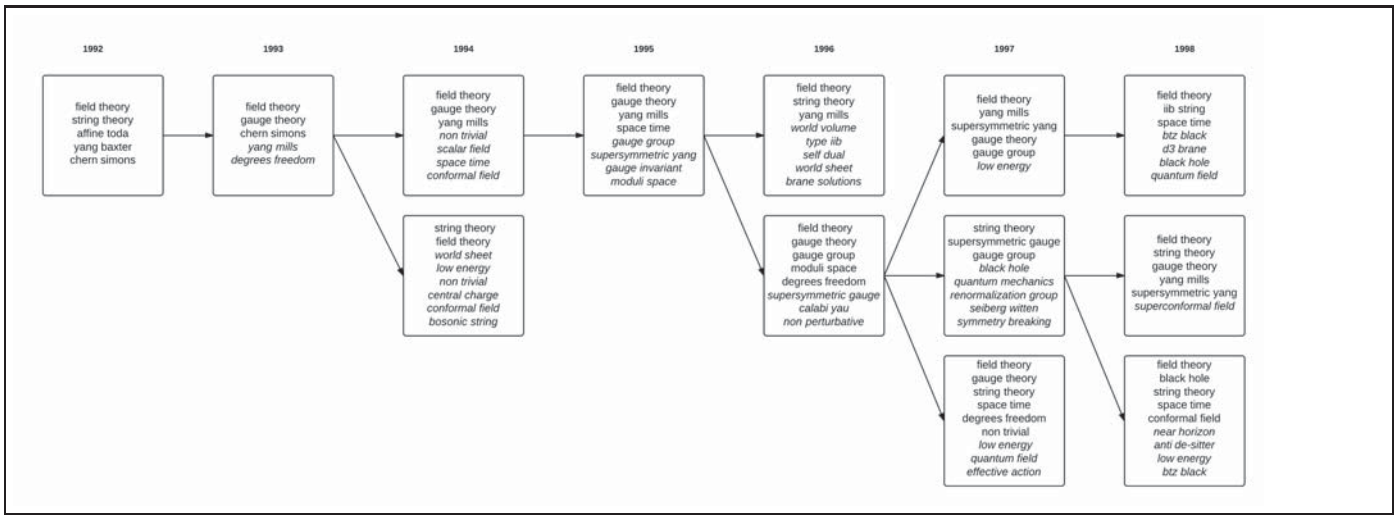


Figure 6: Topic evolution graph for the area *field theory*

Another way to evaluate the correctness of the topic labels extracted is to check if the occurrences of labels coincide with the peaks in their topic strength plots [5]. We have taken the topic strength of a label in a year to be the average of its tf-idf values for documents of that year. Figure 5 shows the topic strength trends of three labels - *Calabi-Yau*, *Chern-Simons* and *field theory*. *Calabi-Yau* occurs only once - in 1996 - and we observe that its topic strength peaks in 1996. Similarly, the strength of *Chern-Simons* theory steadily decreases from 1992 which is reflected by the fact that it is a relevant topic label only in the years 1992 and 1993. However, *field theory* remains relevant throughout the 7-year period and has consistently high topic strength.

Lastly, we mined influential nodes from the citation graph using the PageRank algorithm. Figure 7 compares the distribution of influences and citation counts over the papers. The difference between the two distributions is important in that although papers with high citation counts have larger PageRank scores, citation count is not the only factor in determining the influence. Notice that the first ~15000 papers belonging to the period 1992-1997 have higher PageRank scores on an average than the papers in the 1998-2003 period. This is due to the fact that earlier papers are more likely to be cited by influential papers of subsequent years, thus resulting in higher PageRank scores. One can refer to Figure 7(a) to verify that the citation count distributions of the two periods are similar.

To extract representative papers from each cluster, we used the PageRank scores along with Euclidean similarity to other papers in the cluster with equal weightage to both. One way of evaluating our approach is measuring the extent to which the citation links of representative papers mimic the evolution links of the topic evolution graph. To this end, we measured the minimum similarity between the papers cited by a representative paper in its parent cluster and the representative of the parent cluster, which we call the *nearest citation coefficient*. This tells us how closely the representative papers of a parent-daughter pair of topics are linked by citations. Figure 8 plots the nearest citation coefficients of 64 clusters across 11 years from 1993-2003. We observe that the coefficient values are on the higher side with an average of 0.804. Also, we notice three instances of the coefficient being 1 (a value of 1 indicates a citation between representative papers of parent-daughter clusters).

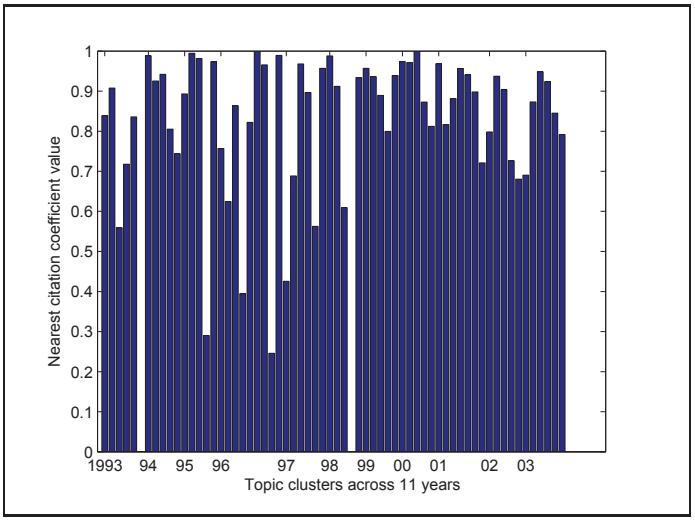
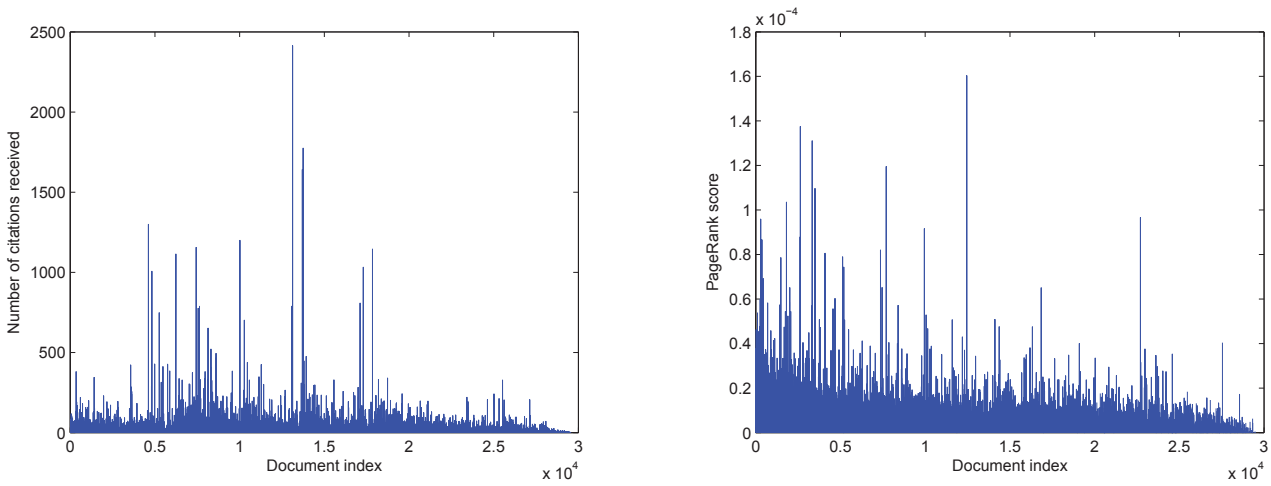


Figure 8: Nearest citation coefficient values for representative papers of topics across 11 years.

VII. Conclusion and Future Work

In this paper, we study the topic evolution problem for a corpus of documents and have proposed a novel, efficient method for generating topic evolution graphs. We implemented our method and observed that it produces accurate results while being time efficient.

Topic detection, topic evolution and mining representative documents were the three areas of focus in our analysis. Considering a time unit of one year, foremost, the documents in each year needed to be grouped based on their topic. To



(a) Number of citations received

(b) PageRank scores

Figure 7: Comparison of citation counts and PageRank scores of documents

this end, we applied SVD on the term-document matrix and then hierarchically clustered the resulting document signatures. For topic detection, we built bigram and trigram frequency matrices and used association to filter meaningless topics. The meaningful n-grams were scored for semantic relevance using first-order relevance measure to choose topic labels for clusters.

Discovering evolution among clusters entailed the analysis of both citation information as well as topic similarity. Our technique to measure the significance of cross citations, along with comparison of topics based on their word distributions, was used for introducing links in the topic evolution graph. Further, we used the PageRank algorithm to mine influential documents and compared them with topics to select representative documents for each cluster.

The dataset we used to run the MATLAB implementation of our technique on was an archive of around 29,000 physics papers published during 1992-2003 in arXiv journal. The results obtained were impressive in terms of correctness and comprehensibility. Importantly, we show the efficiency of our method by observing that the running time is linear in the number of documents and the word count.

Potential avenues for further research include (1) exploring ways to use citation information for topic discovery, on the lines of He et. al. [5], (2) finding a way to eliminate the time unit restriction without compromising on efficiency and (3) exploring the possibility of reducing the running time further by adopting other clustering algorithms while ensuring optimality and cohesion of the resulting clusters.

ACKNOWLEDGEMENT

We would like to extend our thanks to the Dept. of CSE at NITK Surathkal for providing excellent facilities and support for our research. Furthermore, we are grateful to Ms. Saumya Hegde and Dr. Mohit Tahiliani of the Dept. of CSE for their valuable suggestions.

REFERENCES

[1] J. Allan. "Introduction to topic detection and tracking." *Topic detection and tracking*. Springer US, 2002. 1-16.

[2] CY Lin and E. Hovy. "The automated acquisition of topic signatures for text summarization." *Proceedings of the 18th conference on Computational linguistics-Volume 1*. 2000.

[3] T. L. Griffiths, and M. Steyvers. "Finding scientific topics." *Proceedings of the National academy of Sciences of the United States of America* 101.Suppl 1, 2004: 5228-5235.

[4] Y. Jo, C. Lagoze, and C. Lee Giles. "Detecting research topics via the correlation between graphs and texts." *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2007.

[5] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and C. Lee Giles. "Detecting topic evolution in scientific literature: how can citations help?." *Proceedings of the 18th ACM conference on Information and knowledge management.* ACM, 2009.

[6] Y. Jo, J. E. Hopcroft, and C. Lagoze. "The web of topics: discovering the topology of topic evolution in a corpus." *Proceedings of the 20th international conference on World wide web.* ACM, 2011.

[7] S. Morinaga and K. Yamanishi. "Tracking dynamics of topic trends using a finite mixture model." *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2004.

[8] Q. Mei and C. Zhai. "Discovering evolutionary theme patterns from text: an exploration of temporal text mining." *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* ACM, 2005.

[9] R. Schult and M. Spiliopoulou. "Discovering emerging topics in unlabelled text collections." *Advances in Databases and Information Systems.* Springer Berlin Heidelberg, 2006.

[10] D. Zhou, X. Ji, H. Zha, and C. Lee Giles. "Topic evolution and social interactions: how authors effect research." *Proceedings of the 15th ACM international conference on Information and knowledge management.* ACM, 2006.

[11] G. S. Mann, D. Mimno, and A. McCallum. "Bibliometric impact measures leveraging topic analysis." *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on.* IEEE, 2006.

[12] S. Salvador and P. Chan. "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms." *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on.* IEEE, 2004.

[13] K. Seymore and R. Rosenfield. "Large-scale topic detection and language model adaptation." 1997.

[14] A. Barrón-Cedeño and P. Rosso. "On automatic plagiarism detection based on n-grams comparison." *Advances in Information Retrieval.* Springer Berlin Heidelberg. 696-700. 2009

[15] K. W. Church and P. Hanks. "Word association norms, mutual information, and lexicography." *Computational linguistics* 16.1: 22-29. 1990.
[16] Q. Mei, X. Shen, and C. Zhai. "Automatic labeling of multinomial topic models." *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2007.
[17] P. Chen, H. Xie, S. Maslov, and S. Redner. "Finding scientific gems with Google's PageRank algorithm." *Journal of Informetrics* 1.1: 8-15. 2007
[18] L. Katz. "A new status index derived from sociometric analysis." *Psychometrika* 18, 39–43. 1953.
[19] E. Garfield. "Citation analysis as a tool in journal evaluation." *Science* 178, 471– 479. 1972.
[20] L. Page, S. Brin, R. Motwani, & T. Winograd. "The PageRank citation ranking: bringing order to the web." 1999.
[21] J. M. Kleinberg. "Authoritative sources in a hyperlinked environment." *Journal of the ACM (JACM)* 46.5 (1999): 604-632. 1999.
[22] (2003) Datasets for the KDD Cup 2003. [Online]. Available: http://www.cs.cornell.edu/projects/kddcup/datasets.html
[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel et al. "Scikit-learn: Machine learning in Python." *The Journal of Machine Learning Research* 12: 2825-2830. 2011.
[24] C. H. Hubbell. "An input-output approach to clique identification." *Sociometry* 28, 377–399. 1965
[25] D. Zeimpekis and E. Gallopoulos. "TMG: A MATLAB toolbox for generating term-document matrices from text collections." *Grouping multidimensional data.* Springer Berlin Heidelberg, 187-210. 2006.
[26] D. Zhang and Z. Zhu. "A fast approximate algorithm for large-scale latent semantic indexing." *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*. IEEE, 2008.
[27] O. Zamir and O. Etzioni. "Web document clustering: A feasibility demonstration." *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998.