BIOINFORMATICS

Vol. 00 no. 00 2014
Pages 1–8

Biomedical Text Mining in BioMed

Haochen Chen [1,*], Jiaqi Chen [1], Xichan Liu [1] and Chenxi Xin [1*]

[1]Department of Computer Science, Stony Brook University

ABSTRACT

Motivation: BioMed provided us with an open access full-text corpus of over 220000 articles in biomedical research, which could be a good resource for text mining. Firstly, we perform text mining tasks in the corpus, including keywords extraction, static topic extraction and dynamic topic evolution analysis. Secondly, we perform social influence analysis in the authors network, and conduct clustering in the authors citation graph. At last, we combine the results from the first two parts, and analyze topic evolution for specific scientists. The experiment results give us a good overview of biomedical research: we learn about the keyword, topics and topic evolution in biomedical area, the social network structure of the network biomedical researchers, and also the research topic evolution for those highly influntial scientists.

Results: We apply statistal-based methods to the keyword extraction task. We utilize both static and dynamic topic model based on Latent Dirichlet Allocation (LDA) for topic extraction, compare these two models, and find that dynamic model produces a better result when finding the evolution of certain topics. For the authors citation graph, we use several social influence analysis algorithms to discover influential scientists, and prove that unweighted PageRank algorithm gives the best performance. Finally, we choose 2 specific scientists, and conduct dynamic topic modeling to detect their research interest evolution.

Contact: haochen.chen@stonybrook.edu

1 INTRODUCTION

As the volume of biomedical research publication increases with great speed, more and more researchers have a urge need for biomedical text mining tools. According to Cohen and Hersh 2005, some previous work in biomedical text mining focus on the following parts:

Name entity regonition (NER). This task aims at finding name entities in biomedical articles, for example, the name of a specific gene or protein. Comparing to NER in a general corpus, biomedical NER is more challenging because some phrases may correspond to different meanings, and some entities could have different names. As a basic natural language processing task, NER could also be incorporated into a text mining system to enhance the performance of the whole system.

Text classification. In this task, a set of documents and a set of topics (labels) are given, and we want to assign to each document some specific topics which relate to it. Usually this task is semi-supervised or supervised: the labels of some documents are manually annotated, and serve as training data for machine learning algorithms. For example, Dobrokhotov *et al.* 2003 combined NLP techniques with Probabilistic Latent Categoriser (PLR) to categorize biomedical documents. Keerthi *et al.* 2002 and Donaldson *et al.* 2003 use SVM-based algorithms to classify biomedical documents.

But the problem with text classification is that, it needs a lot of manually constructed data: the set of topics, the label of specific documents, etc. Thus, an unsupervised method for extracting documents' topic could be favorable. Blei *et al.* 2003 propose an LDA-based unsupervised method for topic modeling, by estimating topic distribution and then estimating word distribution within each topic. This model has been widely used for topic extraction in various areas. Griffiths and Steyvers 2004 utilizes LDA topic model to discover scientific topics from abstracts from PNAS. Blei and Lafferty 2007 derives a correlated topic model for finding topics in articles from *Science*. But no similar work has been done in a biomedical data set. The only related work in biomedical area is Lin and Wilbur 2007 which uses a probabilistic topic-based model to calculate content similarity (*pmra*), but it aims at finding an effective algorithm for related articles search, while we aim at discovering topics from the articles.

Also, what's interesting to notice is that the BioMed corpus also provides author information and citation information. With these information we could build a huge citation graph among the authors, and performance further social network analysis in it. The only previous work in biomedical social network analysis try to identify interactions among 107 biomedical research scientists (Brieger, 1976). This article uses a social network which is much smaller than the one in BioMed corpus, and it mainly focus on interation between those scientists.

In this article, we finish several text mining and social network analysis tasks based on the BioMed corpus. The Biomed corpus is a full-text corpus published by BioMed Central, which includes 225952 biomedical publications. For the text mining part of our work, we firstly utilize TF-IDF method for keyword extraction in the documents. Then, we apply (static) topic model based on Latent Dirichlet Allocation (Blei *et al.*, 2003) to analyze the topics in the articles of recent 2 years. We also introduce dynamic topic model (Blei and Lafferty, 2006) to analyze the change of topics over time. In addition, topic distribution in the documents are used to analyze hot and cold topics. The result of these experiments give us an overview of not only the hot topics of biomedical research in recent years, but also how the topics changes, which topics emerges and which topics disappear.

Also, we build an authors graph based on the citation data retrieved from the BioMed corpus and perform social network analysis in it. We subsequently conduct basic network analysis in this graph, which includes the degree distribution, betweenness centrality and closeness centrality etc. To find influential scientists in biomedical area, PageRank algorithm and HITS algorithm are

*to whom correspondence should be addressed

utilized. Two most infuential scientists are chosen, and the research topic evolutions of them are analyzed. This helps us understand the change of biomedical research topics from the aspect of individuals. At last, we perform an community detection algorithm based on modmularity maximization (Blondel *et al.*, 2008). Further work could be done to analyze the topic evolution within each community, and the evolution of the communities themselves.

2 METHODS

2.1 Keyword Extraction

Methods for keyword extraction are mainly divided into two categories: statistical-based methods and machine learning-based methods. One of the most widely used statistical-based method is TF-IDF (Salton and McGill, 1983). The most common machine learning method for keyword extraction is SVM. Some previous work regard keyword extraction as a classification problem: all words are labeled with "is a keyword" or "is not a keyword" (Zhang *et al.*, 2006; Richardson and Campbell, 2008). Also, many methods utilize natural language processing techniques to improve experimental results. Hulth 2003 use linguistic knowledge, and Ercan and Cicekli 2007 incorporate lexicon chain to improve their models. Here we utilize statistical methods like TF-IDF and term frequency for keyword extraction, and contrast these two methods.
In TF-IDF, TF and IDF represent term frequency and inverse document frequency of a single word respectively:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $TF_{i,j}$ denotes the ith word's TF value in the jth abstract, $n_{i,j}$ denotes how many times the ith word appear in the jth abstract and $\sum_k n_{k,j}$ denotes the quantity of words in the jth abstract.

$$IDF_i = log\frac{|D|}{|\{j : t_i \in dj\}|}$$

where IDF_i denotes the ith word's IDF value, $|D|$ denotes the total number of abstracts and $|\{j : t_i \in dj\}|$ denotes the number of abstracts that contains the ith word. By using this method, we can get the real keywords which have a high frequency in specific abstracts but have a low frequency in all abstracts.

2.2 Static Topic Extraction

For this part, we extract topics from biomedical articles in recent years based on Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003). LDA is a generative model for topic discovery. In the LDA model, each document contains several topics and the words of a document are generated with certain probability from these topics.
In LDA model, each document is generated as follows:

1. Sampling from the Dirichlet distribution for parameter α and generate the topic distribution θ_i for document i.
2. Sampling from the topic distribution θ_i and generate topic $z_{i,j}$ for j_{th} word of document i.
3. Sampling from the Dirichlet distribution and generate the word distribution $\phi_{z_{i,j}}$ for topic $z_{i,j}$.
4. Sampling from the words distribution $\phi_{z_{i,j}}$ and generate $w_{i,j}$.

Therefore, the total probability of generating a certain document is:

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\phi}; \alpha, \beta) =$$
$$\prod_{i=1}^{K} P(\phi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\phi_{Z_{j,t}})$$

Then, we can get the maximum likelihood estimation as follows:

$$p(w_i|\alpha, \beta) = \int_{\theta_i} \int_{\Phi} \sum_{z_i} p(w_i, z_i, \theta_i, \Phi|\alpha, \beta)$$

Finally, the parameters of model can be estimated by Gibbs sampling with the maximum likelihood estimation.

2.3 Dynamic Topic Extraction

We use the dynamic topic model (Blei and Lafferty, 2006) to analyze the evolution of topics over time. This model is an extension to LDA which can deal with sequential documents with timestamps. Specifically, the documents are grouped by time.
The dynamic process is defined as follow:

$$\beta_{t,k}|\beta_{t-1,k} \sim N(\beta_{t-1,k}, \sigma^2 I)$$
$$\alpha_t|\alpha_{t-1} \sim N(\alpha_{t-1}, \delta^2 I).$$

Where α_t is the document topic distribution at time t and $\beta_{t,k}$ is the word distribution of topic k at time t.
Connect a collection of topic models sequentially and the generative process at time slice t is as follows:

1. Draw topics $\beta_{t,k}|\beta_{t-1,k} \sim N(\beta_{t-1,k}, \sigma^2 I)\forall k$
2. Draw mixture model $\alpha_t|\alpha_{t-1} \sim N(\alpha_{t-1}, \delta^2 I)$
3. For each document:
    1. Draw $\eta_{t,d} \sim N(\alpha_t, a^2 I)$
    2. For each word:
        1. Draw topic $Z_{t,d,n} \sim$ Mult$(\pi(\eta_{t,d}))$
        2. Draw word $W_{t,d,n} \sim$ Mult$(\pi(\beta_{t,Z_{t,d,n}}))$

Where $\pi(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$, $z_{t,d,n}$ is the topic for the n_{th} word in document d in time t, and $w_{t,d,n}$ is the word.

2.4 Hot/Cold Topics Finding

For this part, we utilize the LDA model for dynamic topic extraction to discover hot/cold topics. Suppose we are given the topic-document distribution of choosing a certain topic z_n over a document d as $p(z_n|d)$, then the hotness of a topic in a given time period T is defined as follows:

$$H_{z_n} = \sum_{d \in D} p(z_n|d)$$

where D is the set of documents in T. Suppose we are given the topic-document distribution over a series of time periods $t_1, t_2, \cdots, t_k$, then we could calculate $H_1, H_2, \cdots H_k$ according to the formula above. We build a linear regression model to predict H_{k+1} subsequently:

$$H_i = \beta_i t_i + \epsilon_i$$

Then, we simply use β_i to measure the trend of hotness variation for a certain topic.

2.5 Betweenness Centrality and Closeness Centrality

Both of these two metrics measure the centrality of a certain node in a graph. The betweenness centrality of a node is the number of shortest paths from all vertices to all others which pass through that node. The closeness centrality of a node is the reciprocal of the average distance between that node and all other nodes connected to it.

2.6 Influential Authors Discovery

Our next task is to discover the most influential authors in the authors citation graph. One related area in social influence analysis mainly focus on maximizing the spread of influence (Chen *et al.*, 2009; Kempe *et al.*, 2003; Even-Dar and Shapira, 2007; Chen *et al.*, 2010), while our work is to find

most influential individuals. Another related area is expert finding, which aims at finding persons with expertise for a given topic (Zhang *et al.*, 2007; Balog *et al.*, 2009), but we don't know the specific topics in biomedical research. Thus, here we utilize ranking methods in network to discover the authors with highest rankings. Specificially, unweighted PageRank algorithm (Page *et al.*, 1999), weighted PageRank algorithm (Xing and Ghorbani, 2004) and HITS algorithm (Kleinberg, 1999) are utilized. In the unweighted PageRank algorithm, the PageRank for a given author a_i is:

$$PR(a_i) = (1-d) + d \sum_{a_j \in In(a_i)} PR(a_j)/O_{a_j}$$

where $In(a_i)$ is the set of author which links to a_i, O_{a_j} is the number of outlinks from a_j, d is damping factor and set to 0.85.
In the weighted PageRank algorithm, firstly the popularity from the number of inlinks and outlinks are calculated as follows:

$$W^{in}_{v,u} = \frac{I_u}{\Sigma_{p \in Out(v)} I_p}$$

$$W^{out}_{v,u} = \frac{O_u}{\Sigma_{p \in Out(v)} O_p}$$

where $Out(v)$ is the set of author which v links to, I_u is the number of inlinks of u, O_u is the number of inlinks of u. At last, the weighted PageRank is calculated:

$$PR(a_i) = (1-d) + d \sum_{a_j \in In(a_i)} PR(a_j) W^{in}_{a_j,a_i} W^{out}_{a_j,a_i}$$

In the HITS algorithm, we assign authority value and hub value to each author, and update those values iteratively. For each iteration:

$$\forall a, auth(a) = \Sigma^n_{i=1} hub(i)$$

$$\forall a, hub(a) = \Sigma^n_{i=1} auth(i)$$

where a is any author.

2.7 Community Detection in Citation Graph

For this part, we want to detect the author communities in the citation graph; the communities actually form a partition of the graph. The metric for evaluating the partition is modularity:

$$Q = \sum_{i=1}^{k} (e_{ii} - a_i^2)$$

In which k is the number of communities, e_{ii} denotes the fraction of edges which connect two nodes within the same community i. $a_i = \sum_j e_{ij}$, and denotes the fraction of edges which contain at least one vertex from community i. A higher modularity means a better community partition.
The algorithms for community detection are classified into agglomerative and partitioning methods (Vasudevan *et al.*, 2009). Here we adopt an agglomerative algorithm from Blondel *et al.*. This algorithm assign each vertice to a community in the beginning, and follow two steps to detect communities which maximizes modularity:

1. For each vertex i and each of its neighbor j, we calculate the modularity gain obtained by putting i in j's current community, and putting i into the community of vertex j with largest modularity gain. If we consider the weight of edges, the modularity gain ΔQ by putting i into community C is:

$$\Delta Q = [\frac{\sum_{in} + k_{i,inC} - (\frac{\sum_{tot} + k_i^2}{2m})}{2m}] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2]$$

   Here, $\sum_{in}$ is the weights' sum inside C, $\sum_{tot}$ is the weights' sum of edges connects to C, k_i is the sum of the weights of edges connect to i and m is the weights' sum in the whole graph.

2. Regard the communities found in step 1 as vertices, and build a new graph with them. The weights of edges in the new graph are obtained by the sum of weights in edges between those vertices in the communities.

By iterating through these two steps, we can finally get a community partition of the original graph. It is proven by experiments that this method is very efficient, and its performance (modularity) outperforms other modularity maximization algorithms.

Table 1. Article Number Distribution Over Time

Year Range	69-75	76-80	81-85	86-90	1991-1995
Articles Num	406	683	394	196	261
Year Range	96-00	01-05	06-10	11-12	Total
Num of Articles	244	12144	54584	41878	110790

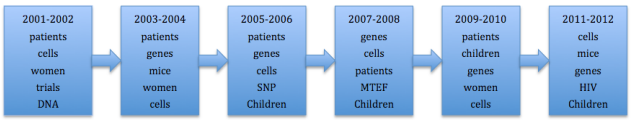

Fig. 1. TF-IDF keyword extraction in 6 time periods.

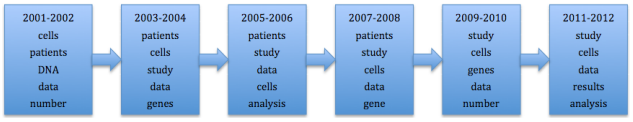

Fig. 2. Word frequency based keyword extraction in 6 time periods.

3 RESULTS

3.1 Data Preprocessing

We use BeautifulSoup in Python for XML data preprocessing. The distribution of number of articles is as of Table 1. We can see that only about 5% articles of the corpus are from 2000 or earlier.

3.2 Keyword Extraction for Articles in Recent Years

We firstly extract keywords in biomedical articles of recent 12 years. We divide the whole time period from 2001 to 2012 into 6 time periods: 2001-2002, 2003-2004, 2005-2006, 2007-2008, 2009-2010 and 2011-2012. For each time period, we randomly choose 1000 sample documents. We implement the TF-IDF algorithm to extract keywords in each time period, and compare the experiment result with keyword extraction method based on word frequency (Figure 1 and Figure 2).

3.3 Topic Extraction for Articles in Recent Years

In this part, we extract biomedical research topics from recent year publications. Due to the limited number of documents before year 2000, we conduct subsequent experiments only in the articles after

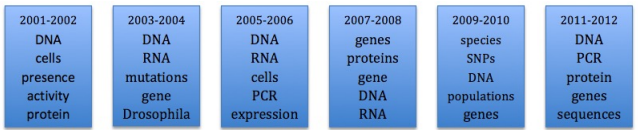

Fig. 3. The top 5 words describing one specific topic among the 6 time periods.

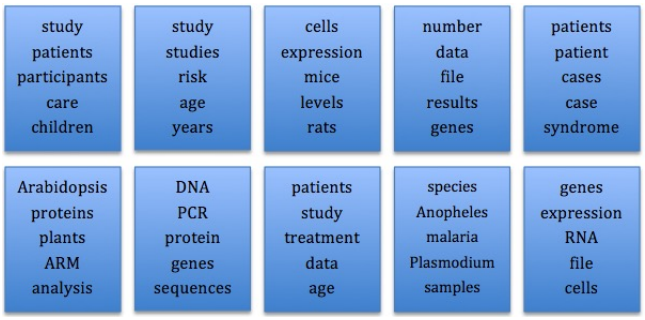

Fig. 4. The 10 topics for time period 2011-2012.

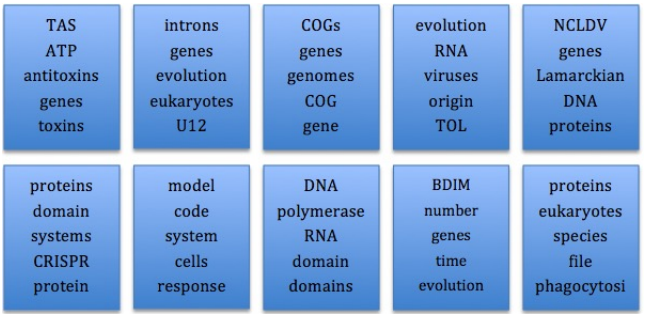

Fig. 5. Research topics of Koonin Eugene.

year 2000. We use the same dataset as in the keyword extraction part.

We extract nouns from the body text of each sample with NLTK (Natural Language Toolkit) for part-of-speech tagging. Then, with the nouns in each document of each year period, we use the GibbsLDA++ toolbox, a C++ implementation of LDA, to discover the topics. Here we set the number of topics to be 10 . Topics of each time period are shown in Figure 3 (Here we only show one same topic out of those ten topics for each time period).

Ten topics of 2011-2012 are shown in Figure 4.

3.4 Topic Extraction for Articles of Certain Authors

Here, we extract the topics of articles of two certain authors (Koonin Eugene and Murray Christopher), who are among the top 10 most influential authors. However, due to the small amount of the two authors' articles, dynamic topic model does not work well. Therefore, we utilize LDA-based static topic model to extract the topics of their articles. The results are as shown in Figure 5 and Figure 6.

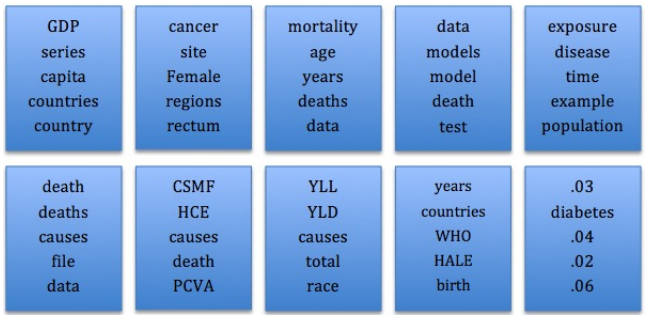

Fig. 6. Research topics of Murray Christopher.

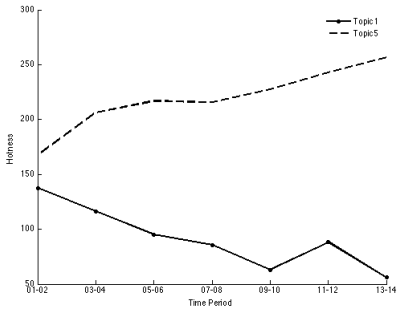

Fig. 7. The hottest topic is topic 5, which is described by keywords $\{sprp1A, RPTPs, Mammographic\}$. The coldest topic is topic 1, which is described by keywords $\{SWL, Decision_Regret_Scale, Brier\}$. This figure shows the trend of hotness for these two topics.

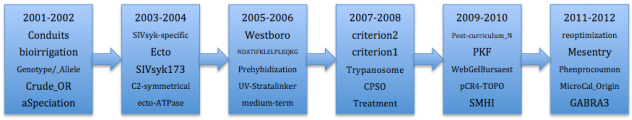

Fig. 8. Topic evolution for topic 1 in 6 time periods.

3.5 Hot/Cold Topics Finding

We use the result from dynamic topic model with year range of 2001-2012. The hottest topic (with the maximum β_i), coldest topic is as of Figure 8. We also predict the hotness of these two topics in year range of 2013-2014

3.6 Dynamic Topic Extraction

In this part, we conduct dynamic topic extraction for biomedical articles. We use dynamic topic model to discover topic evolutions, and the result is shown in Figure **??** and Figure **??**. Also, the result is contrasted with that from the static topic model. We can see from the figure above that the dynamic topic model tends to discover the evolution of topics, and aims at finding very specific keyword within a topic. In opposite, static topic models tends to discover general keywords for a certain topic.

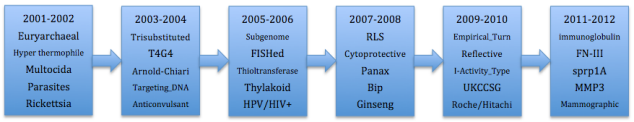

Fig. 9. Topic evolution for topic 2 in 6 time periods.

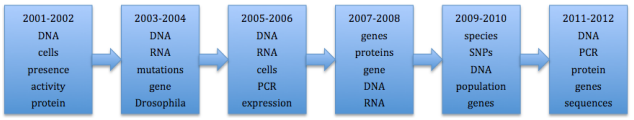

Fig. 10. Topic evolution obtained from static topic model.

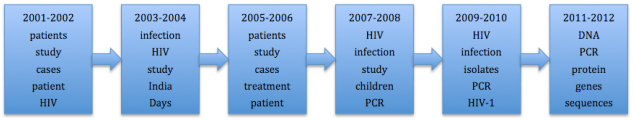

Fig. 11. Topic evolution obtained from static topic model.

Table 2. Citation Graph Statistics

Articles Count	Authors Count
94029	314767
Avg Num of Citations	Avg Num of Citations Within BioMed
29.6	0.53

3.7 Social Network Analysis in Authors' Network

We build a citation graph within the BioMed corpus, in which each vertex is an author, and an edge is added from author a to author b if a's article cites b's article. The basic statistics of the citation graph is as of Table 2:

Notice that most of the citations point to articles NOT within the BioMed corpus, which makes the citation graph to be quite sparse. To make the citation graph more dense and focus on the most influential authors, we eliminate all authors who appears for less than 10 times in the citation graph. Finally we get a citation graph with 39649 vertices and 383011 edges.

3.7.1 Basic Network Analysis Firstly we perform some network analysis tasks with Gephi (Bastian *et al.*, 2009), an analysis and visualization platform for social networks.

Degree Distribution:

The average degree for the citation graph is 9.830, and the in-degree / out-degree distributions are as follows:

Betweenness Centrality and Closeness Centrality Distribution: Table 100 shows the betweenness centrality and closeness centrality distribution of the citation graph: The distribution for betweenness and closeness in the citation graph are as Figure 13:

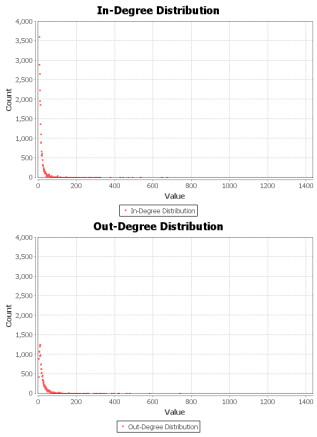

Fig. 12. Indegree Distribution and Outdegree Distribution of the Citation Graph

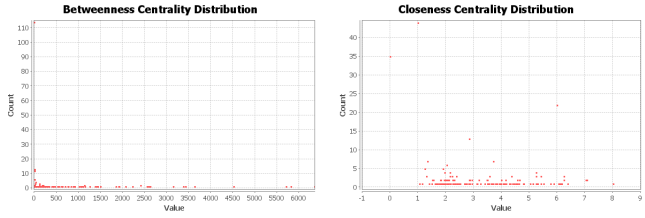

Fig. 13. Betweenness Centrality Distribution and Closeness Centrality Distribution of the Citation Graph

Table 3. Average Citation Count and H-index for the 3 Algorithms

	uw-PR	w-PR	HITS
Citation Count	12117	10952	3858
H-index	46	35	28

3.7.2 Influential Authors Discovery For the unweighted PageRank (uw-PR), weighted PageRank(w-PR) and HITS algorithm, we run 1000 iterations over all the vertices. We search for total number of citations and h-index for the authors from Web of Science database manually, as the evalution for these 3 algorithms. The average number of citations and h-index for the top 20 scientists get from the 3 algorithms are in Table 3:

From the table we can see the performance of the unweighted PageRank algorithm is slighter better than the weighted PageRank algorithm, and both of the PageRank algorithms' performance are much better than that of the HITS algorithm. Also, all the 3 algorithms successfully discover many experts with extremely high citations and h-index. The top 10 scientists we discover with unweighted PageRank algorithm are in Table 4.

Table 4. Scientists with Topic 10 PageRank

Name	Sum of the Times Cited	H-index
Koonin Eugene	56056	124
Noble Paul	9014	46
Meltzer Eric	198	5
Grimshaw Jeremy	14965	50
Rambaut Andrew	18810	52
Edgar Robert	13851	24
Marincola Francesco	15861	62
Murray Christopher	38414	87
van Mechelen Willem	8080	46
Kim Harold	5046	27

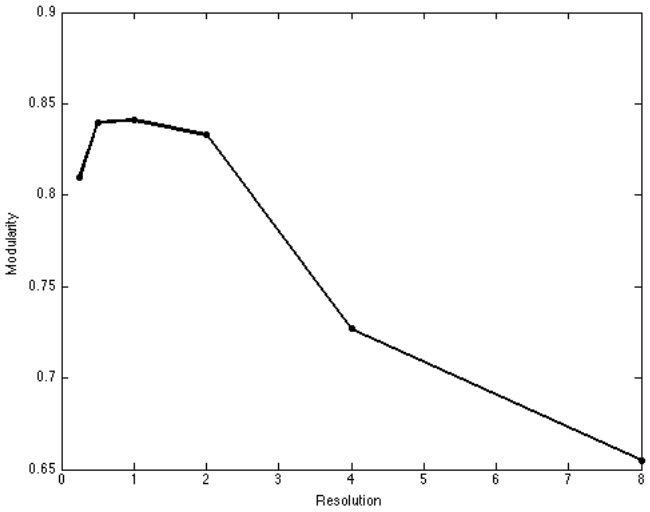

Fig. 14. This figure describe the relationship between resolution and modularity. We get the highest modularity value with an resolution of 1.0.

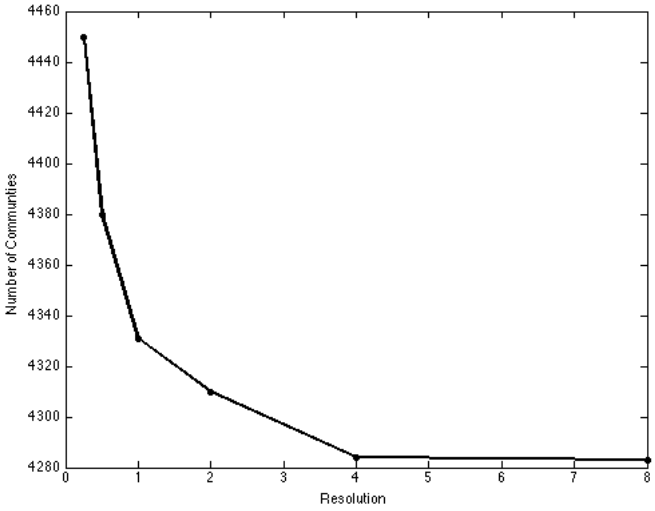

Fig. 15. This figure describe the relationship between resolution and the number of communities. The number of communities decreases when resolution increases.

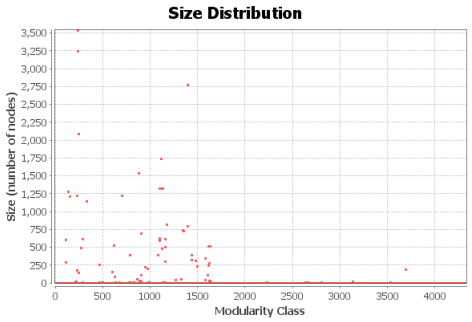

Fig. 16. Communities size distribution with resolution = 1.0. The edges in the graph are limited to citations within the BioMed corpus, which means we omit a large portion of real edges. This also results in a large number of communities: most of the communities only consist of very few authors. We only have 4 communities with size bigger than 2000.

3.8 Community Detection in Citation Graph

In this part, we use modularity maximization algorithm to perform graph clustering. One important parameter for the modularity maximization algorithm is the resolution limit. With a low resolution, the algorithm tends to get a large number of small communities, while with a high resolution, the algorithm tends get a small number of large communities. (Lancichinetti and Fortunato, 2011; Lambiotte *et al.*, 2008). Firstly we conduct an experiment to discover the relationship between resolution value and modularity, number of communities as Figure 14 and Figure 15. Thus, here we choose the resolution value of 1.0 for the modularity maximization algorithm. The distribution of community size we get are in Figure 16.

Finally, we visualize the citation graph with Gephi (Figure 17):

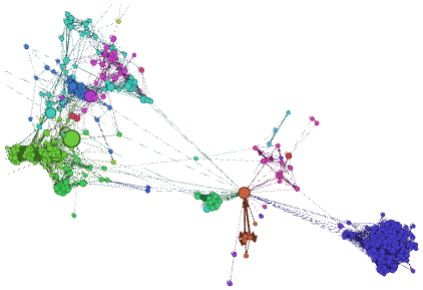

Fig. 17. Community Visualization for the Citation Graph. The size of vertices is determined by their PageRank, while the color is determined by their belonging community.

4 DISCUSSION

We come across several problems with BioMed data processing efficiency and the data itself. XML processing is very time-consuming, and our solution is to preprocess XML data, and store some basic information (full-text, author info, etc.) in plain text files. The distribution of data is uneven (shown in Table 1), with over 95% of the articles are published after year 2000. Also, some documents in the document lack necessary information for experiments: many documents before year 2000 don't have body text. In practice, we simply throw away early years with very few documents, and focus on the documents after year 2000.

For the keyword extraction part, obviously TF-IDF derives a much better result than the naive frequency counting method. But although top words we get from TF-IDF do describe some potential topics in biomedical articles, we can see those keywords are still not satisfying, because usually a topic cannot be described accurately by a single keyword.

Thus, we introduce LDA-based topic model here, which provides an unsupervised method to discover possible topics within the corpus. Here, each topic is described by a set of words. An additional benefit for the topic model is that, as we could obtain the distribution of topics over documents with LDA, we could estimate the popularity of each topic by refering to this distribution. Moreover, we introduce time dimension to topic model, and build a dynamic topic model in the BioMed corpus. This allows us to further understand the appearance and disappearance of biomedical topics over time. One shortcome with topic model is: it could be very memory-consuming and time-consuming when we have a large corpus. We resolve this problem by random sampling from the original corpus.

Also, we note that text preprocessing with natural language processing techniques could significantly improve the performance of subsequent text mining tasks. Here we simply use NLTK to extract nouns from text, as the topics and keywords should be described by nouns. Some future work could be utilizing biomedical NER tools to these text, and do disambiguation for the name entities. We could also assign these name entities with higher weight when conducting keyword/topic extraction.

For the social network analysis part, we finish basic analysis of the authors citation network, and apply ranking algorithms to rank the influence of biomedical scientists. Also, community detection in the citation network show us the diversity of biomedical research. This community partition results could be used for topic evolution in specific biomedical research topics.

5 CONCLUSION

In this paper, we perform various text mining and social network analysis tasks based on the BioMed corpus. Firstly we extract keywords from the corpus, and further give the evolution of keywords in biomedical area over time. As we find that a single keyword alone cannot represent a certain topic, we turn to unsupervised topic modeling methods, in which a number of topics are given, and each topic is described by a set of keywords. Firstly, a static topic model based on LDA is utilized to generate topics of general interest in recent biomedical research. What's more, we can identify hot and cold topics from the theta distribution in LDA topic model. Then we use a dynamic topic model which incorporate the "timestamp" of research articles, to discover topics evolution over time. These automatically generated topics can give people a better understanding of recent biomedical research, and even predict the popularity of topics by analyzing the evolution of previous research topics.

We also conduct series of social network analysis tasks in the large-scale authors network, which include basic network analysis, influential authors discovery and community detection. We discover the most influential biomedical scientists, and further analyze research topic evolution for them. These experiment result let people understand biomedical research from the aspect of researchers, in contrast to the aspect of publications. An interesting future work in this part is to detect the evolution of communities in biomedical research, which may involves some dynamic community detection algorithms (Tantipathananandh *et al.*, 2007). Also, get topic evolution for each research community is also interesting, which could let us know the topic evolution for specific areas of biomedical research.

REFERENCES

Balog, K., Azzopardi, L., and de Rijke, M. (2009). A language modeling framework for expert finding. *Information Processing & Management*, **45**(1), 1–19.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks.

Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM.

Blei, D. M. and Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, **3**, 993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008.

Brieger, R. L. (1976). Career attributes and network structure: A blockmodel study of a biomedical research specialty. *American sociological review*, pages 117–135.

Chen, W., Wang, Y., and Yang, S. (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM.

Chen, W., Wang, C., and Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM.

Cohen, A. M. and Hersh, W. R. (2005). A survey of current work in biomedical text mining. *Briefings in bioinformatics*, **6**(1), 57–71.

Dobrokhotov, P. B., Goutte, C., Veuthey, A.-L., and Gaussier, E. (2003). Combining nlp and probabilistic categorisation for document and term selection for swiss-prot medical annotation. *Bioinformatics*, **19**(suppl 1), i91–i94.

Donaldson, I., Martin, J., De Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K., *et al.* (2003). Prebind and textomy–mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC bioinformatics*, **4**(1), 11.

Ercan, G. and Cicekli, I. (2007). Using lexical chains for keyword extraction. *Information Processing & Management*, **43**(6), 1705–1714.

Even-Dar, E. and Shapira, A. (2007). A note on maximizing the spread of influence in social networks. In *Internet and Network Economics*, pages 281–286. Springer.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, **101**(Suppl 1), 5228–5235.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics.

Keerthi, S. S., Ong, C. J., Siah, K. B., Lim, D. B., Chu, W., Shi, M., Edwin, D. S., Menon, R., Shen, L., Lim, J. Y., *et al.* (2002). A machine learning approach for the curation of biomedical literature: Kdd cup 2002 (task 1). *ACM SIGKDD Explorations Newsletter*, **4**(2), 93–94.

Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international*

conference on Knowledge discovery and data mining, pages 137–146. ACM.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, **46**(5), 604–632.

Lambiotte, R., Delvenne, J.-C., and Barahona, M. (2008). Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*.

Lancichinetti, A. and Fortunato, S. (2011). Limits of modularity maximization in community detection. *Physical Review E*, **84**(6), 066122.

Lin, J. and Wilbur, W. J. (2007). Pubmed related articles: a probabilistic topic-based model for content similarity. *BMC bioinformatics*, **8**(1), 423.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web.

Richardson, F. and Campbell, W. M. (2008). Discriminative keyword selection using support vector machines. In *Advances in Neural Information Processing Systems*, pages 209–216.

Salton, G. and McGill, M. J. (1983). Introduction to modern information retrieval.

Tantipathananandh, C., Berger-Wolf, T., and Kempe, D. (2007). A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726. ACM.

Vasudevan, M., Balakrishnan, H., and Deo, N. (2009). Community discovery algorithms: an overview. *Congressus Numerantium*, **196**, 127–142.

Xing, W. and Ghorbani, A. (2004). Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 305–314. IEEE.

Zhang, J., Tang, J., and Li, J. (2007). Expert finding in a social network. In *Advances in Databases: Concepts, Systems and Applications*, pages 1066–1069. Springer.

Zhang, K., Xu, H., Tang, J., and Li, J. (2006). Keyword extraction using support vector machine. In *Advances in Web-Age Information Management*, pages 85–96. Springer.