Predicting Concept-based Research Trends with Rhetorical Framing

Jifan Yu[1], Liangming Pan[2], Juanzi Li[1], and Xiaoping Du[3]

[1] Tsinghua University, Beijing, China
{yujf18.mail,lijuanzi}@tsinghua.edu.cn
[2] National University of Singapore, Singapore
e0272310@u.nus.edu
[3] Beihang University, Beijing, China
xpdu@buaa.edu.cn

Abstract. Applying data mining techniques to help researchers discover, understand, and predict research trends is a highly beneficial but challenging task. The existing researches mainly use topics extracted from literatures as objects to build predicting model. To get more accurate results, we use concepts instead of topics constructing a model to predict their rise and fall trends, considering the rhetorical characteristics of them. The experimental results based on ACL1965-2017 literature dataset show the clues of the scientific trends can be found in the rhetorical distribution of concepts. After adding the relevant concepts' information, the predict model's accuracy rate can be significantly improved, compared to the prior topic-based algorithm.

Keywords: Scientific Trends Analysis · Concept Extraction · Scientific Discourse Analysis.

1 Introduction

With the rapid growth of research community, it becomes increasingly difficult for researchers to see the complete picture of how a research field has been evolving, with the overwhelming amount of scientific literatures. Therefore, applying data mining techniques to help researchers discover, understand, and predict research trends becomes a highly beneficial but challenging task.

Most of the existing methods for research trends analysis are topic-based [2, 4, **?**]. They first extract research topics using topic modeling on a collection of scientific papers, and then study the rise and fall of each topic based on machine learning or statistical methods. Among these methods, Vinodkumar et al [12]. predicts trends of a topic based on the changing of its rhetorical role. The rhetorical role of a topic is the purpose or role it plays in the paper: such as background, research objectives, methods, conclusions, etc. Rhetorical functions that topics take part in serve as strong clues of the topic evolution. For example, if a topic was often to be employed as a method in the past, but was mentioned a lot as background recently may signal an increase in its maturity and a possible decrease in its popularity.

However, topic-based methods can not provide the level of granularity needed to support an in-depth analysis of research dynamics in scientific literature, due to the following reasons. First, a topic is a word distribution, which requires further human interpretation to understand its meaning. Many word distributions are noisy and hard to interpret. Second, topics are often too coarse-grained. For example, the topic about SVM may contain the algorithm of SVM, the applications of SVM as well as other SVM-based methods.

In this paper, we propose to analyze the research trends of scientific documents from a different perspective, i.e., concept-based analysis. Concepts in scientific papers are key phrases that express the main idea of the paper, for example, problems (e.g., NER), techniques (e.g., SVM), domains (e.g., machine translation), datasets (e.g., Semeval 2010), and evaluation metrics (e.g, F1 Score). Detecting the rise and fall of concepts rather than topics can provide a more fine-grained view of research dynamics. We follow the idea of Vinodkumar et al. to investigate how the change of rhetorical functions influences the rise and fall of scientific concepts. However, performing concept-based analysis poses several unique challenges. First, concepts in scientific papers are hard to be identified because most of them are newly-proposed and domain-specific. Second, the significant variability of expressions makes a concept often has many identical mentions (e.g., SVM and support vector machine). If we regard each mention as a separate concept, it will raise the data sparsity issue. However, clustering identical concept mentions is tough, because lexical similarity is not reliable for the clustering, and traditional clustering algorithms are hard to generate tight enough clusters.

To address the above challenges, we introduce a novel algorithm that uses the rhetorical structural features of the concept to analyze the rise and fall of scientific concepts. Specifically, we first propose a method to extract the concepts from the scientific literature and rationally merge the synonyms of them to obtain the concept data set. Then, based on the work of Vinodkumar et al [12]., we automatically annotate each sentence of the abstract with a rhetorical role. Finally, based on different rhetorical roles of the concept at different times, we analyze the evolution trajectory of the concepts. The entire study was conducted on the ACL1965-2017 dataset consisting of 36,929 actual papers. The experimental results show that the rhetorical functions of the concept and the changing trends of its related concepts have significant effects on the growth or decline of its popularity in the future.

Contributions: The three main contributions of our paper are: 1) we proposed a new method to extract concepts from scientific literature and merge the identical concept mentions; 2) we show that the rhetorical function distribution of a concept also reflects its temporal trajectory, and that it is predictive of whether the concept will eventually rise or fall in popularity; 3) we significantly improve the prediction accuracy of the existing model by considering related concepts.

2 Related works

Our work is based on the work of keyphrase extraction and scientific trend analysis. Keyphrase extraction provides research candidates to form concepts, while scientific trend analysis with scientific discourse analysis provides research ideas for this work.

2.1 Keyphrase Extraction

Keyphrases are defined as a set of terms in a document that give a brief summary of its content for readers. Automatic keyphrase extraction is widely used in information retrieval and digital library [17, 25]. Keyphrase extraction is also an essential step in various tasks of natural language processing such as document categorization, clustering and summarization. There are two principled approaches to extracting keyphrases: supervised and unsupervised. In the unsupervised approach, graph-based ranking methods are state-of-the-art [16, 26]. These methods first build a word graph according to word co-occurrences within the document, and then use random walk techniques to measure word importance. After that, top ranked words are selected as keyphrases.

The supervised approach [21] regards keyphrase extraction as a classification task, in which a model is trained to determine whether a candidate phrase is a keyphrase. Our work chooses this approach and groups identical key phrases to build the concept dataset.

2.2 Scientific Trends Analysis

In literature metrology and scientometrics, there are a lot of researches on the trends of scientific research. Research methods can be broadly divided into two types, one focusing on the citation of the literature and one focusing on textual information. The former researchers often used topological methods to identify those emerging research topics in advance from the common reference clustering [7, 10] or mutual reference networks of the literature [6]. The other part starts with the text of the paper itself. For example, Mane and Guo use the word burst to find new and emerging scientific fields [8, 9], while Small makes emotional assessments of the various cited texts and shows the different potential of scientific terms [11].

Vinodkumar's work [12] analyzes traditional scientific trend using scientific discourse analysis [24]. By dividing the literature abstracts into rhetorical functions, statistics on the frequency of occurrence of each scientific topic in different regions are analyzed, and finally contribute to predicting the trend of it. Our work gain the idea from it and optimize the model by changing research objects to concepts.

3 Methods

To provide an end-to-end solution of predicting the trends of scientific concepts, we design a two-stage framework. First, we automatically extract concept mentions from scientific papers, and then we propose a novel clustering algorithm to merge identical or similar concept mentions. Second, based on the work of Vinodkumar et al., we design a model that uses the rhetorical features to predict the rise and fall of scientific concepts. In the following sections, we will introduce the two parts in details.

3.1 Discovery of Scientific Concepts

Given a collection of scientific papers, we first extract concept mentions and design a novel algorithm to cluster identical concepts. The concept mentions are clustered in suitable granularity to address the data sparsity issue, and facilitate the trend prediction in the next stage.

Extracting Concept Mentions We design a three-stage method to automatically extract scientific concepts from the literature as follows.

1. We first extract candidate concept mentions using linguistic patterns.
2. We then calculate the feature vector for each candidate mention.
3. Finally, we classify each candidate mention as valid scientific mention or not by training a binary classification model.

Step 1 Considering that most concepts are noun phrases [1], we obtain candidate course concepts by extracting all noun phrases in the paper using the following POS pattern, where JJ is presented as adjective, NN as noun and IN as preposition.

$$\left\{ \left(\langle JJ \rangle^{*} \langle NN^{*}_{.} \rangle + \langle IN \rangle \right) ? \, \langle JJ \rangle^{*} \langle NN^{*}_{.} \rangle + \right\} \tag{1}$$

Step 2 In order to filter out noise from the candidates, and obtain qualified concept mentions, we train a binary classifier using features using different aspects of information. Specifically, we calculate the feature vector for each candidate, and then train a binary classifier to determine whether a candidate is a valid concept mention. The feature design is shown in the Table 1.

Table 1. Feature Engineering of Noun-phrases.

Category	Feature	Description
Frequency-based Features	Term Frequency	The frequency of the term in the paper
	Sentence IDF	The percentage of sentences in which the term appears
	PMI	Point Mutual Information of the term in this paper
Statistical Features	Term Length	The length of the term
	Lexical Cohesion	Lexical Cohesion for term t is
	Max Word Length	The length of the longest word
Grammatical Features	Is acronym	Whether the term is an acronym
	Is capital	Whether the first letter of this term is capital
	Is named entity	Whether the term is a named entity
Positional Features	First Occurrence	Normalized positions of first occurrence
	Last Occurrence	Normalized positions of last occurrence
	Spread	Difference between first occurrence and last occurrence
	In title	Whether the term appears in the paper title
	In abstract	Whether the term appears in the paper abstract

Step 3 After feature engineering, we train a binary classifier to classify candidate as concept mention or not concept mention.

Clustering Identical Concept Mentions Unlike topics, the number of extracted concepts can be very big, which results in many infrequent concepts. Performing trend analysis on these infrequent concepts will raise the data sparsity problem. Therefore, after obtaining all concept mentions from the collection of scientific papers, we propose a mention clustering algorithm to automatically group identical and similar concept mentions. Different from traditional clustering algorithms (e.g., K-Means), our algorithm clusters similar concept mentions without specifying the cluster numbers, which is more suitable in our problem setting.

Before introducing the algorithm, we first propose two assumptions about the nature of identical concept mentions, based on observations on real-life scientific literature.

- **Co-occurrence**: Identical mentions tend to co-occur within close context windows.
- **Co-reference**: Different mentions of a same concept are likely to cite the same paper.

Figure 1 uses the concept "Word Embedding" as an example to show these two properties of identical mentions in actual texts. Authors use the abbreviation and synonym like "WEs", "vector representations of words", etc. to emphasize and explain this concept, and these all cite a same milestone paper.

Dense real-valued distributed representations of words known as **word embeddings** (**WEs**) (***Mikolov et al., 2013***) have become ubiquitous in NLP…

Recently, the **distributed representations of words** (***Mikolov et al., 2013***) have been shown that …

… optimizes **vector representations of words** (**word embeddings**) (***Mikolov et al., 2013***) such that they can predict other context words occurring in a small window…

…and measuring the similarity with **word2vector** (***Mikolov et al., 2013***).

Fig. 1. Identical mentions often co-occur as explanations and cite same papers.

Based on the above assumptions, we could identify identical or similar concepts by capturing these two aspects of information, i.e., co-occurrence and co-reference information.

- We capture co-occurrence information by learning **concept embeddings**[23]. Word Embeddings aims to maximum the probability of the context words given the center word in a small context window, so we first replace each concept mention as a single token in the corpus, then train word-embeddings on it to obtain the semantic representations for each concept mention. The cosine distance between two mention vectors, denoted as $sim\left(m_i, m_j\right)$, can represent their semantic relatedness.

- We capture co-reference information by extracting **citing sentences**. We define a citing sentence as the sentence which contains at least one citation. For a citing sentence containing a concept mention m and cites a paper p, we obtain a mention-paper pair (m, p) and finally get the set of cited papers $cit\left(m\right)$ for each mention m.

Based on the above components, we propose the **Concept Mention Grouping Algorithm**. The algorithm iteratively merges similar concepts, and makes sure the following property holds throughout all the iterations. For concept mentions in the same cluster, the semantic relatedness between any two mentions must be larger than σ, and all of the mentions in the cluster must at least co-reference θ papers. The details of the algorithm is shown in Algorithm 1.

For a cluster c, we define $cit\left(c\right) = \bigcap_{m \in c} cit\left(m\right)$. We also define *cluster similarity* as $sim\left(c_i, c_j\right) = \min\left\{sim\left(m_1, m_2\right)\right\}$, when $m_1 \in c_i$ and $m_2 \in c_j$.

Algorithm 1 Concept Mention Grouping Algorithm.

Input: Concept mentions as $M = \{m_1, \cdots, m_n\}$ and Cited Papers for each mention as $Cit\left(M\right) = \left\{cit\left(m_1\right), \cdots, cit\left(m_n\right)\right\}$.

Output: Concepts as $C = \{c_1, \cdots, c_k\}$, where $c_i = \left\{m_{i_1}, \cdots, m_{i_{|c_i|}}\right\}$, $c_1 \bigcup \cdots \bigcup c_k = M$ and $c_i \bigcap c_j = \emptyset$, $\forall c_i, c_j \in C$.

1: Initialize concepts $C = \{c_1, \cdots, c_n\}$, $c_i = \{m_i\}$;
2: Ranking by $sim\left(m_i, m_j\right)$ to obtain a list of tuples, where $\forall k$, $i_k \neq j_k$, $\forall k_1 < k_2$, $sim\left(m_{i_{k_1}}, m_{i_{k_1}}\right) \geq sim\left(m_{i_{k_2}}, m_{i_{k_2}}\right)$, $L = \left\{\left(i_1, j_1, sim(m_{i_1}, m_{j_1})\right), \cdots, \left(i_{n^2-n}, j_{n^2-n}, sim(m_{i_{n^2-n}}, m_{j_{n^2-n}})\right)\right\}$;
3: Pop the first tuple of L. $\left(i, j, sim\left(m_i, m_j\right)\right) = Pop\left(L\right)$;
4: **if** $sim\left(m_i, m_j\right) < \delta$ **then return** C, end
5: **end if**
6: Find m_i and m_j currently belong to which concept, denoted as c_{m_i} and c_{m_j}.;
7: Deleting some weak classifiers in E_n so as to keep the capacity of E_n;
8: **if** $sim\left(c_{m_i}, c_{m_j}\right) == sim\left(m_i, m_j\right), cit\left(c_{m_i}\right) \bigcap cit\left(c_{m_j}\right) \geq \theta$ **then**
9: $\quad C = Merge\left(c_{m_i}, c_{m_j}\right)$
10: **else**
11: $\quad$ GOTO **step 3**
12: **end if**

It can be proven that for any cluster c_i that Algorithm 1 outputs, we have $\forall m_h, m_l \in c_i, sim\left(m_h, m_l\right) \geq \sigma$ and $\left|\bigcap_{m \in c_i} cit\left(m\right)\right| \geq \theta$. This ensures that concept mentions with high co-occurrences and co-references are clustered together. As for the time complexity, Step 2 is of $O\left(n^2 log\left(n\right)\right)$, Step 3-11 is of $O\left(n^2\right)$, so the total time complexity is $O\left(n^2 log\left(n\right)\right)$.

3.2 Predicting the rise and fall of Scientific Concepts

Rhetorical Feature Extraction and Modeling To setup the rhetorical analysis, we first use a tool based on *CRF* named *ArgZoneTagger* [12] to place each sentence in abstract with seven different tags, which expresses its rhetorical attributes.

The seven tags are: BACKGROUND(The scientific context), OBJECTIVE(The specific goal), DATA(The empirical dimension used), DESIGN(The experimental setup), METHOD(Means used to achieve the goal), RESULT(What was found) and CONCLUSION(What was inferred). LDP are seven features corresponding

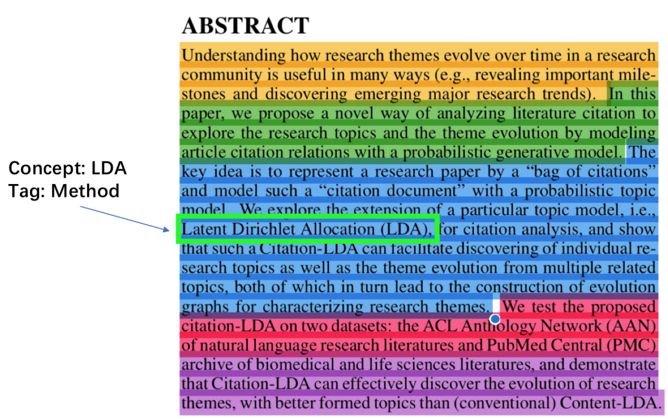

Fig. 2. A tagged abstract and concept "LDA" occurs in "Method" segment.

to the percentage of concepts across the seven rhetorical function labels (e.g., % of time the concept is a METHOD like Figure 2), which can be calculated as below. $LD\,(c, s_i, t)$ is the frequentness Concept c occurs in segment s_i during time t.

$$LDP\,(c, s_i, t) = LD\,(c, s_i, t) \,/ \sum LD\,(c, s, t) \tag{2}$$

Digging further, we can also find other 7 features named LDR, reflecting how much a concept's LDP changed from former period to now.

$$LDR\,(c, s, t_i) = LDP\,(c, s, t_i) - LDP\,(c, s, t_{i-1}) \tag{3}$$

We also can combine LDP and LDR features to build a more completed 14-dimension feature named LDS, the experiment will test the behavior of these three features. Next, we simplified the task of predicting lifting into a classification problem. By calculating the difference between the heat in the next period of time and the current, we divide the concepts into concepts of ascent, concepts of decline, and concepts of stability. This work can be done by training a classifier based on L2 logistic regression.

Using related concepts to optimize the model In fact, the lifting of a concept not only depends on itself. A rising concept may cause changes in its associated multiple concepts as topics do [20]. Therefore, it is also meaningful to investigate the trend of a few concepts related to a specific concept. Thanks to our discovering work, we can also use the similarity of two concepts as $sim\left(c_i, c_j\right)$ in Section 3.1 to find out the most related ones of concept c_i. Adding the recent trends of them as assistant features may also give a rise of the accuracy of our prediction.

4 Experiments

4.1 Results of Discovering Scientific Concepts

According to the process of finding the concept, we first extract all qualified noun phrases from the literature and conduct preliminary screening to obtain the following results, for Noun-phrases within 5-grams:

- Number of extracted candidate concept mentions: **21,308,601**
- Number of **unique** candidate concept mentions: **5,962,170**
- Number of candidate mentions **per paper**: **577**

As what is mentioned in section 3.1, we build a binary classifier to classify candidate as concept mention or not concept mention using Sem-Eval 2010 as our training data. It contains 144 ACL papers with human-annotated key phrases. For each paper, we use the feature vectors of human-annotated key phrases as

Table 2. Behavior of different classifiers.

Classifier	Precision	Recall	F1-Score
K-Nearest Neighbors (K=3)	70.12	65.65	67.81
L2 logistic	65.23	64.16	64.69
L1 logistic	75.51	62.57	68.43
Linear SVC	72.47	66.66	69.44
RBF SVM	72.44	65.94	69.04
Navie Bayesian	71.83	61.63	66.34
Decision Tree	76.55	63.14	69.20
Ada Boost	69.87	68.98	69.42
Random Forest	77.87	59.80	67.65

positive examples, and randomly sample some candidate key phrases as negative examples. It results a total of 3683 training examples, with 1624 positive examples and 2059 negative examples. In order to select the most suitable classification algorithm, we tried almost all the mainstream algorithms and used 5-fold cross validation to evaluate the results of different classifiers. The results are displayed in detail in the Table 2.

In fact, the performance of mainstream algorithms is not much different, so we chose **Linear SVC** with the highest F1-Score as the classification algorithm, because it can also calculate the score of each phrase in the article to facilitate future work. Finally we get **13,002** unique concept mentions with three conditions:

1. The score given by the mention classifier must be **positive**.
2. For each paper, we only select its **top-20 scoring** candidate mentions.
3. The candidate must appears in **at least 5 papers**.

Since we have raised the **Concept Mention Grouping Algorithm** with the feature of citing reference, we first extract all citing sentences in the ACL Dataset resulting in below:

– Resulted in a **total of 538,956** citing sentences.
– With an **average of 14.59** citing sentences per paper.
– Number of **unique** cited papers is: **111,453**.

The thresholds of the semantic relatedness between any of a concept cluster's mentions, σ and the number of co-reference papers, θ can be adjusted to get different result of clustering work as Table 3. We select the better-grained case

Table 3. Results of Concept Mention Grouping Algorithm

θ	σ	Number of clusters	Average mentions per cluster
3	0.6	8,326	1.56
3	0.7	9,136	1.42
1	0.6	1,730	7.51
1	0.7	4,557	2.85

of $\theta = 1$, $\sigma = 0.7$ as a concept data set, and Table 4 shows a few examples in this case.

Table 4. Examples of concept clusters($\theta = 1$, $\sigma = 0.7$)

Concept	Mentions	Concept	Mentions
5	metadata information meta-information meta information metadata	49	morphosyntactic annotation part-of-speech annotation morphological annotation
7	inter-annotator kappa statistic inter-annotation agreementintra-annotator agreement inter-annotator agreement inter-coder agreement annotator agreement inter-annotator agreement scores inter-annotator agreements kappa coefficient inter-annotator agreement inter-rater agreement	18	morphosyntactic tags part-of-speech tag part-ofspeech tags part-of-speech part-of-speech tags partof-speech tags morphological tag partof-speech tag morphological tags part-of-speech labels morphological attributes part of speech parts-of-speech pos tags

4.2 Results of Predicting the rise and fall of Concepts

Our task comes to be building a predicting model based on Concepts' rhetorical features. We first divide the ACL Dataset which contains 36,929 articles into 369 subsets by time order. Each subset keeps 100 articles in average and represents a specific historical period's research overview.

Tracking concept popularities In order to train the prediction model, we need to find out the actual popularity-change information of the concept as ground truth. Counting the frequency of concepts as their popularity is a common method in this area of research. Therefore, we count the popularity of the concept in each period and calculate the $trend\,(c,t)$ as trend of the concept. $trend\,(c,t)$ is defined as a three-valued quantity of 0,1,2. reflecting the state that $popularity\,(c,t_i)$ is equal to, larger than, and smaller than $popularity\,(c,t_{i+1})$. And $popularity\,(c,t)$ can be calculated as below.

$$popularity\,(c,t) = PaperwithConcpet\,(c)\,/SumOfPaper\,(t) \tag{4}$$

Start the rhetorical predict model We use ArgZoneTagger to give each sentence in abstracts with 7 labels. Matched to the tagged abstracts, each concept presents a LDP feature in every period as Section 3.2 shows. Simultaneously, LDR and LDS feature can be calculated, so we finally find 1,676,976 terms of data for each feature. After pairing with the *trend* data, we randomly select 75% of them as training data and the remaining 25% as test data, using L2 logistic regression(C=10) to construct the prediction model. We also train a topic model on same dataset of ACL like Vinodkumar's prior work to give a comparison. The results of predict model are shown in Table 5. The results show that using LDS

Table 5. The result of predict model using rhetorical features.

Feature of Concept	Accuracy	Feature of Topic	Accuracy
LDP	62.1%	LDP	60.3%
LDR	70.1%	LDR	70.8%
LDS	74.3%	LDS	72.0%

feature gives best accuracy, while concepts instead of topics perform more validities of predicting the rise and fall of scientific trends, due to a better granularity and semantic concentration.

Using related concepts to optimize the model Since the similarity of two concepts $sim\,(c_i,c_j)$ can be calculated, we select a concept's nearest related concepts' *trend* information in the former period as an assistant. We combine different number of related concepts and the accuracy changes like Table 6.

Table 6. The results of combined features

Test ID	Selected Features	Accuracy
1	Nearest 7 concepts' recent trends	53.12%
2	LDP + Nearest 7 concepts' recent trends	66.44%
3	LDR + Nearest 7 concepts' recent trends	72.32%
4	LDS + Nearest 7 concepts' recent trends	81.21%
5	LDS + Nearest 8 concepts' recent trends	81.17%
6	LDS + Nearest 10 concepts' recent trends	80.10%
7	LDS + Nearest 14 concepts' recent trends	77.15%
8	LDS + Nearest 4 concepts' recent trends	80.82%
9	LDS + Nearest 6 concepts' recent trends	81.06%

The result indicates the help of related concepts and the difference between the numbers of related concepts involved. Test 1 tells the related concepts' information can only be an assistant. Test 2-4 tells LDP, LDR and LDS's accuracy is also like the results without new features. Test 4-10 shows that 7 related concepts seem to be the best to give the model an optimization, resulting in a 9% rise of accuracy compared with prior predict algorithm.

5 Conclusion and Future work

In this paper, we propose a novel method to predict the rise-and-fall trends from the scientific concepts' rhetorical features. First we extract concepts from scientific literature and merge the identical concept mentions as research objects, then use ArgZoneTagger tool tagging sentences with 7 labels to discover rhetorical role of the concept in them. We calculate a concept's LDP, LDR, LDS feature to build the predict model and also consider the information of related concepts to optimize it. In experiments, we run the concept extractor in ACL1965-2017 dataset, resulting in 13,002 key phrases and 4,557 concept clusters. Besides, test shows changing the objects to concepts instead of topic gives a rise of predicting accuracy, while the model can perform better to with related concepts' recent trends.

Considering further, we can use the concepts dataset and their rhetorical features to analyze and discover more clues of scientific trends using other algorithm like RNN, which we leave as our future directions.

Acknowledgement

The work is supported by National Key Research and Development Program of China (2017YFB1002101), NSFC key project (U1736204, 61661146007) and THUNUS NExT Co-Lab.

References

1. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. Conference on Empirical Methods in Natural Language Processing (pp.366-376). Association for Computational Linguistics.(2010)
2. Griffiths, T. L., & Steyvers, M.: Finding scientific topics. Proc Natl Acad Sci U S A, 101 Suppl **1**(1), 5228-5235.(2004)
3. PAAnderson, A., Dan, M. F., & Dan, J.: Towards a Computational History of the ACL: 19802008. Acl-2012 Special Workshop on Rediscovering 50 Years of Discoveries (pp.13-21). (2013)
4. Blei, D. M., & Lafferty, J. D.: Dynamic topic models. Proc. International Conference on Machine Learning (pp.113-120).(2006)
5. Blei, D. M., Ng, A. Y., & Jordan, M. I.: Latent dirichlet allocation. J Machine Learning Research Archive, 3, 993-1022.(2003)
6. Shibata, N., Kajikawa, Y., Takeda, Y., & Matsushima, K.: Detecting emerging research fronts based on topological measures in citation networks of scientific publications. Technovation, 28(11), 758-775.(2008)

7. Shibata, N., Kajikawa, Y., Takeda, Y., & Matsushima, K.: Comparative study on methods of detecting research fronts using different types of citation. Journal of the Association for Information Science & Technology, 60(3), 571580.(2009)
8. Mane, K. K., & BRner, K.: Mapping topics and topic bursts in pnas. Proc Natl Acad Sci U S A, 101(Suppl 1), 5287-5290. (2004)
9. Guo, H., Weingart, S., & Brner, K.: Mixed-indicators model for identifying emerging research areas. Scientometrics, 89(1), 421-435.(2011)
10. Small, H.: Tracking and predicting growth areas in science. Scientometrics, 68(3), 595-610. (2006)
11. Small, H.: Interpreting maps of science using citation context sentiments: a preliminary investigation. Scientometrics, 87(2), 373-388. (2011)
12. Prabhakaran, V., Hamilton, W. L., Dan, M. F., & Dan, J.: Predicting the Rise and Fall of Scientific Topics from Trends in their Rhetorical Framing. Meeting of the Association for Computational Linguistics (pp.1170-1180).(2016)
13. Grineva, M., Grinev, M., & Lizorkin, D.: Extracting key terms from noisy and multitheme documents. International Conference on World Wide Web, WWW 2009, Madrid, Spain, April (pp.661-670). DBLP.(2009)
14. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. Conference on Empirical Methods in Natural Language Processing (pp.216-223). Association for Computational Linguistics.(2003)
15. Litvak, M., & Last, M.: Graph-based keyword extraction for single-document summarization. Mmies 08 Workshop on Multi-source Multilingual Information Extraction & Summar (Vol.64, pp.17–24).(2008)
16. Liu, Z., Li, P., Zheng, Y., & Sun, M.: Clustering to find exemplar terms for keyphrase extraction. Conf Empirical Methods in Natural Language Processing, 1, 257-266.(2009)
17. Turney, P. D.: Learning algorithms for keyphrase extraction. Information Retrieval, 2(4), 303-336.(2000)
18. Wan, X., & Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. National Conference on Artificial Intelligence (pp.855-860). AAAI Press.(2008)
19. Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., & Zheng, X., et al.: Lightlda:big topic models on modest computer clusters. 1351-1361.(2014)
20. Wang, X., & Mccallum, A.: Topics over time:a non-Markov continuous-time model of topical trends. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp.424-433). ACM.(2006)
21. Turney, P. D.: Learning to extract keyphrases from text. , cs.lg/0212013(cs.LG/0212013).(2002)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111-3119.(2013)
23. Teufel, S.: Argumentative zoning: information extraction from scientific text.(1999)
24. Liakata, M.: Zones of conceptualisation in scientific papers: a window to negative and speculative statements. The Workshop on Negation and Speculation in Natural Language Processing (pp.1-4). Association for Computational Linguistics.(2010)
25. Nguyen, T. D., & Kan, M. Y.: Keyphrase Extraction in Scientific Publications. International Conference on Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers (Vol.4822, pp.317-326). Springer-Verlag.(2007)
26. Mihalcea, R.: Textrank : bringing order into texts. Emnlp, 404-411.(2004)