Leveraging the Citation Graph to Recommend Keywords

Ido Blank, Lior Rokach, Guy Shani
Information Systems Engineering &
Deutsche Telekom Innovation Laboratories
Ben Gurion University, Israel
{blank,liorrk,shanigu}@bgu.ac.il

ABSTRACT

Users of scientific papers databases, such as CiteSeer^X, Google Scholar, and Microsoft Academic, often search for papers using a set of keywords. Unfortunately, many authors avoid listing sufficient keywords for their papers. As such, these applications may need to automatically associate good descriptive keywords with papers. This is a well-studied problem given the complete text of the paper, but in many cases, due to copyright privileges, research papers databases do not have the complete text, only metadata, such as the title and abstract. On the other hand, research papers databases typically maintain the citation network of each paper. In this paper we study the problem of predicting which keywords are appropriate for a scientific paper, using only the citation network. We compare our method with predicting keywords using the title and abstract, concluding that the citation network provides much better predictions.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Keywords Recommendation, Citation Graph, Academic Papers

1. INTRODUCTION

Searching online for scientific papers is a common task for every modern scientist. There are a number of search engines, such as CiteSeer^X, Microsoft Academic, and Google Scholar, that offer services including searching for relevant papers, and viewing the paper metadata — its title, authors names and affiliations, its abstract, and the papers that it cites. The most important part of the paper, its textual content, however, is often unavailable for downloading directly from the search engine due to copyright privileges. In these cases the search engine typically forwards the user to a webpage where the user can download or buy the paper.

RecSys'13, October 12–16, 2013, Hong Kong, China.

When searching for papers in a specific area, keywords can provide a significant advantage in rapidly identifying relevant papers. Such keywords are manually added by the authors of a paper, attempting to encapsulate important aspects of the paper, such as its main research area, the modeling approach that was chosen, the methods that were used, or specific algorithms that were leveraged. There is currently no agreed upon method for choosing keywords, and authors tend to follow different methodologies in choosing them. As such, keywords tend to be highly diverse and noisy, and in many cases contain spelling mistakes. Some authors even misunderstand the essence of the keyword concept. Furthermore, many authors avoid the effort of writing keywords altogether, resulting in papers with no keywords (in the KDD CUP 2013 challenge, about 75% of the papers donated by Microsoft Academic have no keywords).

It is hence advantageous for the search engine to automatically identify keywords for a paper when none exist, or even add additional keywords when number of keywords is insufficient. This task is typically known as keyword recommendation. Most of the work on keyword recommendation for documents in general and scientific papers in particular is to use the document text, perhaps as a bag-of-words, and some technique for identifying important terms within the text, such as the well known TF-IDF approach. In the case of scientific papers, due to the aforementioned copyright problem, the complete text of the paper is unavailable to the search engine. The metadata, i.e., the title, authors, abstract, venue and references, are usually available even though these sections are also protected through copyright. However, practically many publishers (such as ACM or IEEE) provide access to this information without the need to purchase the paper or sign-in.

In this paper we focus on the problem of recommending keywords in the case where the full text of the paper is unavailable. We examine the use of both the matadata and the citation graph in identifying relevant keywords. We show that the upper bound on the value of the information in the citation graph is about twice as much as the upper bound on the information in the title and abstract. Furthermore, we show that simple recommendation techniques that rely on the citation graph provide much better precision-recall curves than using the abstract and title. The proposed method can help search engines to improve their performance and to authors to select the most suitable keywords for their papers.

Our experiments use the huge CiteSeer^X database. We discuss problems with the keywords that were automatically

Source	Method	Corpus	Domain	Predictive Performance	Paper
Full-Text	Supervised Learning	161 papers	Physics	precision@5=55.4%	[3]
Full-Text	Statistical	1 book	Genetics	recall@283=40% and precision@283= 10%	[5]
Full-Text	Statistical	1 book	Genetics	precision@k for k<50 is above 60%	[1]
Full-Text	Statistical	2 books	Varied	NA	[8]
Full-Text	Supervised Learning	332 papers	CS	True-positive of 73%, True-negative of 79%	[11]
Full-Text	Supervised Learning	250 papers	CS	precision@5= 31%, recall@5=10%	[7]
Abstract	Supervised Learning	2000 papers	CS	precision@10=29.7%	[6]
Abstract	Supervised Learning	80 papers	CS	precision@9=23%	[2]

Table 1: Text-based methods for keywords extractions in academic corpus.

extracted from scientific papers by the CiteSeer^X team, and explain the procedure that we ran to clean up the keywords.

2. RELATED WORK

One obvious source for keywords recommendation is the article's text. Shah et al. [10] show that the abstract is the most important section in the text for obtaining qualitative keywords, but analysis of the other sections is required for obtaining a high recall. Several researchers tried to employ text-based methods that analyze the full-text. Specifically, extraction of keyphrases from text is a well-known task that has already been examined in the academic domain. Fewer methods extract keywords merely from the title and the abstract [6, 2].

Although the above mentioned keywords extraction methods can be used for keywords recommendation, no citation-based method for keywords recommendation has been reported in the literature. Most of the methods above have been tested on relatively small corpus containing a few hundreds of papers. The objective of this research is to quantitatively examine the usefulness of citation graph for keywords recommendation on a large corpus. Citation graph analysis has been used in other tasks such as citation recommendation [4] and researchers ranking [9] and have shown to significantly improve the predictive performance. Thus, we hypothesize that citation analysis can also be useful in keywords recommendation

3. SCIENTIFIC PAPERS DATASET

We experiment with the CiteSeer^X dataset, consisting of $1,945,157$ papers with metadata, including the paper's title, its abstract, papers that it cites, and its keywords, if they exist. As much of the CiteSeer^X collection process is automated, the dataset contains much noise. We now discuss the clean up process that we used in order to reduce the noise in the dataset. In all the phases below we use standard stemming techniques to compare words.

First, we removed $194,963$ papers that did not have a reasonable title (e.g. only two characters). We ignore below the ACM "Terms", which we consider to be too generic to be useful as keywords.

The automated keyword extraction mechanism often fails to properly identify the keywords section, and in many cases the extracted keywords contain obvious parsing noise, such as adding the "Keyword", the name of the keywords section, as a keyword. In many papers the automated extraction also added the words "Abstract", "Terms", "Additional Key Words and Phrases", and so forth. All these were removed from the set of keywords.

We search for prepositions and conjunctions such as "and" or "the", and remove them from the keywords. When a conjunction appears within a keyword, the keyword is split into two. For example, when encountering the keyword "data-mining and clustering", we would split it into two keywords — "data-mining" and "clustering".

A common problem when allowing people to manually add keywords are different terms with the same meaning. Consider for example the terms "learning algorithm", "statistical learning", and "machine learning", all denoting the same meaning. We need to recognize that these terms are all identical. To resolve this problem we use the downloadable Wikipedia database[1], containing a table listing which search terms get redirected to other search terms. Whenever we identify a keyword that Wikipedia redirects to another keyword, we replace this keyword with the redirect target, which we call the keyword's standard form.

Some authors chose long keywords composed of several terms, e.g. "matched multiscale asymptotic analysis". It is unlikely that this keyword would appear more than once or will be searched for in this exact pattern, so it is desierable that the search engine will be able to acknowledge its various parts. We split keywords containing more than 3 words, and appearing less than 5 times in our database using the following technique — we search for popular keywords (keywords that appear more than 50 times in our database) that appear as substrings of the given unpopular keyword, giving priority to longer popular keywords. In addition for popular keywords, we also search for Wikipedia terms that appear as substrings in long keywords. For example, the keyword "matched multiscale asymptotic analysis" will be split into "asymptotic analysis", "multiscale", and "matching". In this process we did not use over-generic keywords such as "algorithm", "data", and "modeling".

Following this process we remain with $470,064$ papers with keywords containing $2,075,639$ keywords. There are $233,797$ distinct keywords and Figure 2 shows the distribution of the number of appearances of keywords.

4. RECOMMENDING KEYWORDS

We now explain two methods for recommending keywords for a given paper, when the complete text of the paper is unavailable. We first present a method based on the citation graph structure, assuming that cited papers have similar keywords. We then present a more standard baseline method, based on the title and the abstract of the paper

4.1 Citation Graph Recommendations

Our first method uses the keywords of papers cited by the current paper (child papers) and by papers that cite at least one paper that the current paper does (sibling papers). We look for keywords that appear in these two sets of papers and construct a list of recommended keywords for the current paper (Figure 1).

[1] `http://wiki.dbpedia.org/Downloads38`

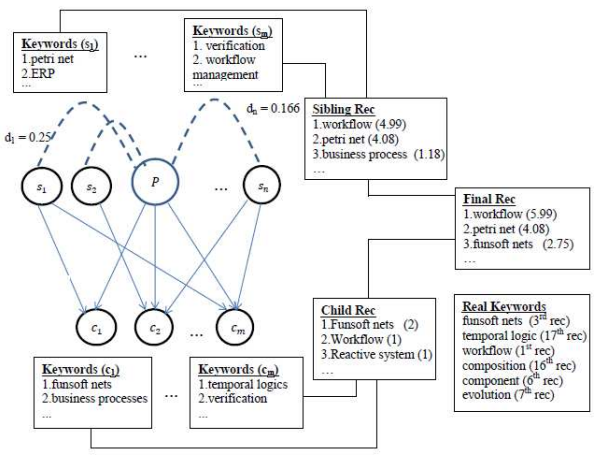


Figure 1: Citation network based example.

Siblings share at least one child with the current paper. We give higher weight to siblings that share more children with the current paper, using the Jaccard similarity: $Jaccard(p_i, p_j) = \frac{citations(p_i) \cap citations(p_j)}{citations(p_i) \cup citations(p_j)}$ where p_i and p_j are papers, and $citations(p)$ are the papers that p cites.

Let $c_{p,k}$ denote the number of times that keyword k appears in a child paper of the current paper p. Let $s_{p,k} = \sum_{p_i \in siblings(p), k \in p_i} Jaccard(p, p_i)$ be the sum of Jaccard weights of siblings of p where k is used as a keyword. We order the keywords that appear in children and siblings of p by decreasing $c_{p,k} + s_{p,k}$.

We use the Wikipedia redirects to identify identical keywords that are written differently in various papers.

4.2 Text-based Recommendation

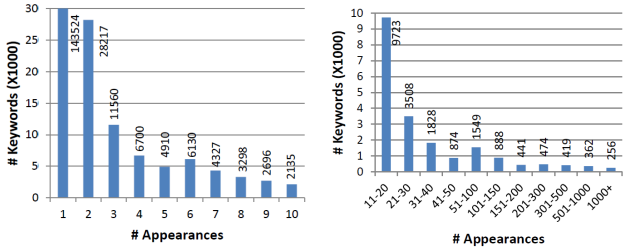
We now list three more orthodox methods that we used for recommending keywords based on the abstract and title only, which will be used as the baseline for comparison.

TF-IDF: Perhaps the most popular method for finding keywords for a document is to look for terms that appear in the document text. Then, these terms can be ordered in a ranked list, for example by their TF-IDF score. In our case we do not have the full text of the paper, but we still have its title and abstract. We hence implemented a standard TF-IDF algorithm, where the candidate terms are only keywords that appear in our dataset.

In many cases, a keyword can appear as a substring of another keyword. For example, the keyword "clustering algorithm" contains the keyword "algorithm". In such cases, we give priority to longer keywords, and ignore the substring keyword. In addition, when identifying keywords in the text, we use standard stemming techniques, and also use the Wikipedia redirects to search for synonyms of keywords.

Yahoo!: Yahoo! Content Analysis is a free service that can also extract keywords from text, detecting entities/concepts within unstructured content. We used the Yahoo! API, providing the title and the abstract and received a list of detected concepts ranked by their overall relevance and their corresponding pages in Wikipedia if applicable.

Location-based: We implemented a novel algorithm that leverages the location of the keyword in the title and abstract. For each keyword that appears in the title and abstract we give a score based on where it appears.



(a) 10 times or less. (b) More than 10 times.

Figure 2: The distribution of keywords appearances in our dataset.

The title generally attempts to define the paper in as little words as possible. Therefore, each keyword that appears in the title gets 1 point. The abstract typically begins with a more general explanation and then becomes more technical. We therefore give higher score to keywords that appear near the beginning of the abstract.

Let $l_{p,k}$ be the location, i.e., the number of characters from the beginning of the abstract of paper p to the first appearance of keyword k, and let $|a_p|$ be the number of characters in the abstract. Let $a_{p,k}$ be the number of times that keyword k appears in the abstract of paper p. The score of k for the abstract of p is: $a_{p,k} \cdot \left(1 - \frac{l_{p,k}}{|a_p|}\right)$. The total score of keyword k for paper p is its abstract score, with an addition of 1 if it also appears in the title.

In our dataset the average length of the title is about 10 words, and the average length of the abstract is 138.7 words.

5. EMPIRICAL EVALUATION

We evaluate the two methods above over the CiteSeer^X datatbase. We follow standard accuracy evaluation by picking a paper with known keywords, computing a list of recommended keywords for it, and comparing the recommended list to the true list of keywords of that paper, computing the precision and recall of the recommendation list.

As we explained above, many of the papers in the database did not have any keywords associated with them, and thus cannot be used in this type of evaluation. Furthermore, our citation-based method requires that not only the paper under evaluation would have keywords, but also that its children and siblings have keywords, further reducing the number of evaluation candidate papers. In our experiments we used only papers where at least 10 of their children have keywords. There were 37,848 such papers.

Figure 3 shows the results of various versions of our algorithms on this datatset. The TF-IDF method over the abstract and title achieved very low results, with maximal precision of 0.05 and maximal recall of 0.07, and was removed from the graph.

As we can see, the citations method, which uses both the children and the siblings of the current paper, achieves the best results. To better understand this performance, we use two auxiliary methods, one that uses only the children, and one that uses only the siblings. Here, the siblings clearly add more value than the children. This is not very surprising, because the number of siblings is much higher than the number of children. A paper typically cites 10-20 papers directly, but can have hundreds of siblings.

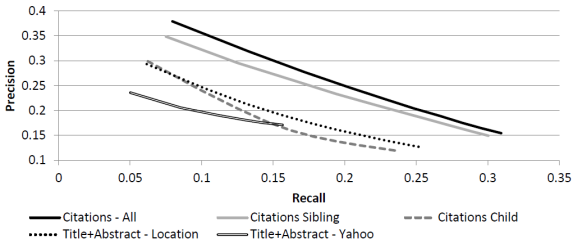


Figure 3: Precision-Recall curves for various keyword recommendation methods, with recommendation lists ranging from 1 to 10. Differences are statistically significant, except for the child citations and the abstract location before the 3rd keyword.

The method that looks at the location of keywords in the abstract and title is doing as well as the children only method, but not as good as the method that uses the siblings or the method that uses both the siblings and the children. This demonstrates the power of using the citation graph over using only the paper metadata for finding good keywords. The location-based method does better than the mature Yahoo! service, which is rather surprising. This can be because the Yahoo! service is generic and designed for any document type, while we leverage several properties specific to scientific papers, such as the structure of the title and abstract.

While the results above seem promising, it may well be that our text-based methods that use the title and abstract are simply not sophisticated enough to provide good results. We hence analyze the paper's keywords appearance in the various sources. In the reduced dataset that was used in the experiments, we find that only 55% of the keywords appear in the abstract and title, while 65% of the keywords appear in the children and the siblings. These numbers provide an upper bound on the possible recall using these two sources. Moreover, almost all keywords that were found in the abstract and title, were also found in the citation graph. This means that if we optimally use the citation graph, there are almost no additional keywords in the abstract and title. Still, the title and abstract may be leveraged to rank the keywords.

6. CONCLUSION

In this paper we explore the problem of recommending keywords for scientific papers using the citation graph. We demonstrate that using simple methods over papers cited by the current paper, and papers that cite the same papers as the current paper, we can get good recommendations in terms of precision and recall, compared with using the title and the abstract of the paper. Obtaining keywords using citation graph can contribute to two aspects. First, previous works show that in order to get reasonable performance using keyword extraction methods, the full-text should be available. The availability of the full-text is not always guaranteed due to many reasons such as copyright restrictions or the author's reluctance to share the text before it is been published. Citation-based method relaxes these restrictions. Moreover, citation-based method can be used in conjunction with text-based methods, to augment the predictive performance of the system.

We also show that there is still much potential in the citation graph for additional improvements, which we will explore in future work. We also intend to combine the citation graph and the abstract and title methods, to better rank the keywords. Another potential extension is in identifying clusters of related keywords, that are interchangeable.

Acknowledgements

We would like to thank Lee Giles and Prasenjit Mitra from The Pennsylvania State University for kindly providing us with the CiteSeerX data used in our research.

7. REFERENCES

[1] C. Carretero-Campos, P. Bernaola-Galván, A. Coronado, and P. Carpena. Improving statistical keyword detection in short texts: Entropic and clustering approaches. *Physica A: Statistical Mechanics and its Applications*, 2012.

[2] Y. HaCohen-Kerner. Automatic extraction of keywords from abstracts. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 843–849. Springer, 2003.

[3] Y. HaCohen-Kerner, Z. Gross, and A. Masa. Automatic extraction and learning of keyphrases from scientific articles. In *Computational Linguistics and Intelligent Text Processing*, pages 657–669. Springer, 2005.

[4] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430. ACM, 2010.

[5] J. P. Herrera and P. A. Pury. Statistical keyword detection in literary corpora. *The European Physical Journal B*, 63(1):135–146, 2008.

[6] A. Hulth. 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, 2003.

[7] S. N. Kim and M.-Y. Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 9–16. Association for Computational Linguistics, 2009.

[8] M. Ortuno, P. Carpena, P. Bernaola-Galván, E. Munoz, and A. Somoza. Keyword detection in natural languages and dna. *EPL (Europhysics Letters)*, 57(5):759, 2002.

[9] L. Rokach, M. Kalech, I. Blank, and R. Stern. Who is going to win the next association for the advancement of artificial intelligence fellowship award? evaluating researchers by mining bibliographic data. *Journal of the American Society for Information Science and Technology*, 62(12):2456–2470, 2011.

[10] P. K. Shah, C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Information extraction from full text scientific articles: Where are the keywords? *BMC bioinformatics*, 4(1):20, 2003.

[11] C. Wu, M. Marchese, J. Jiang, A. Ivanyukovich, and Y. Liang. Machine learning-based keywords extraction for scientific literature. *Journal of Universal Computer Science*, 13(10):1471–1483, 2007.