Academic Web Search Engine — Generating a Survey Automatically

Ye Wang, Zhihua Geng, Sheng Huang, Xiaoling Wang, Aoying Zhou

Department of Computer Science
Fudan University
220 Handan Road, Shanghai 200433, China

{wangyee, zhihuageng, shhuang, wxling, ayzhou}@fudan.edu.cn

ABSTRACT

Given a document repository, search engine is very helpful to retrieve information. Currently, vertical search is a hot topic, and Google Scholar [4] is an example for academic search. However, most vertical search engines only return the flat ranked list without an efficient result exhibition for given users. We study this problem and designed a vertical search engine prototype Dolphin, where the flexible user-oriented templates can be defined and the survey-like results are presented according to the template.

Categories and Subject Descriptors

H.3.6 [INFORMATION SYSTEMS]: Library Automation—Large text archives; H.3.4 [INFORMATION SYSTEMS]: Systems and Software—Distributed systems;

General Terms: Algorithms, Design, Documentation

Keywords: Vertical Search engine, Academic, Model, Web

1. INTRODUCTION

Academic search engine, such as Google Scholar, becomes an interesting and promising topic in recent years. However most search engines return the results to users by a list and users must scan each item/webpage one by one in order to collect and re-organize these information based on users' requirements. It is often difficult to find useful information from thousands or hundreds of WebPages for tenderfoots or junior students. This procedure is time-consuming even for sophisticate.

It is a routine for students/researchers to compose a survey on the corresponding research subject. But, a researcher, who is not familiar with the knowledge of the domain, tends to spend most of his/her time wandering on the Internet, searching for documents, downloading and studying these documents, looking for references, and then searching and downloading again.

In this poster, we study how to organize the result list in order to provide a clear and useful vision for information requestors. Though some previous works have studied some clustering methods for search engine, we pay attention to some further topics here: (1) searching strategy over XML-based documents collection; (2) result representation formats, including template-based information organization, hierarchical information clustering, graphic representation, etc. XML is a de-facto standard for information representation and exchange, and most academic data, such as DBLP dataset and CiteSeer dataset. Thus, searching over XML document collections is a promising topic. In this system, we use XML as the underlying data model. Based on XML data model, we study searching and indexing methods for

efficient academic search and provide a survey-like template to organize extracted information for users.

Generally speaking, a typical computer science survey includes the following components. ◆What is the definition of this research problem? ◆How many categories the previous work can be divided into? ◆Last decade, what are the trends about this area, and which direction is the most promising? ◆Who is the major player of this area, and which is the dominant one among his papers? ◆What is the distribution of the published papers of this area in recent years? ……

To meet the academic survey requirements listed above, we developed an academic web search engine system—Dolphin. For given query keywords, Dolphin returns the definition of the subject, the categories of the existing work, some statistical data, etc. And the most important is that all results are calculated and organized online. Our system provides a useful tool for analyzing and organizing information automatically.

The rest of this paper is organized as follows. Section 2 introduces the system architecture and components. Some snapshots of our running system are presented in Section 3, followed by our conclusion in Section 4.

2. SYSTEM DESIGN

2.1 System Architecture

Most of the data that Dolphin system used is collected from web and represented by XML. Currently, in our document collection, there are more than 700000 papers, published during the period from 1970 to 2006.

The system architecture in Figure 1 shows that there are five modules: (1) the module of Focused Crawler, which is responsible for collecting data; (2) Data storage module; (3) the module of Data Center; (4) Query Processor and (5) Graphics UI module.

In the following subsections, these components and underlying methods are described one by one.

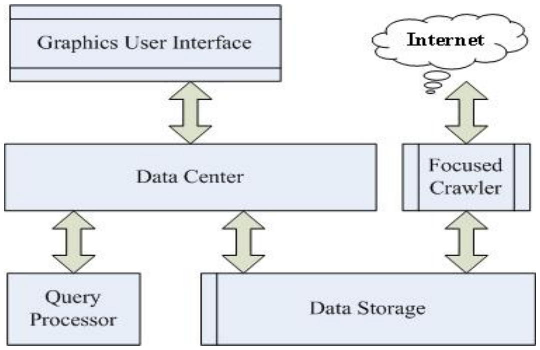


Figure 1. System architecture of Dolphin

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

2.2 Focused Crawler Module

Our system highly depends on the topic related data, hence we use topic focused crawler to collect data. A novel score function is presented to evaluate the URLs' correlation about the specific topic. Only the URL whose score is greater than a given threshold is fetched. Three factors contribute to the score. The first one is the content of given web pages, including title, keyword, text and description. The second one is the anchor text of the URL and the third one is the link structure of the connected URLs and pages. For those satisfied Web pages, we access them, analyze the content inside, organize them with XML format, and store them into data storage. Figure 2 shows the hierarchical structure of the data collecting module and the procedure for data parsing.

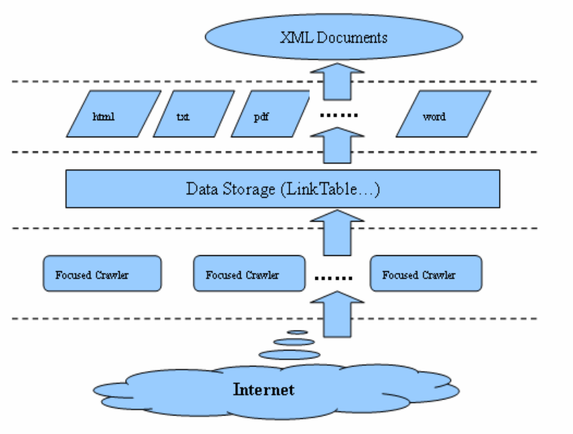


Figure 2. Hierarchy of the data collecting module

2.3 Data Storage Module

We design seven types of xml files: Paper Document, Book Document, Thesis Document, WWW Document, Organization Document, People Document, and Project Document. And we use Lucene[2] to index xml files.

2.4 Data Center Module

The Module of data center receives the users' queries and then parses them to get related documents from the storage component. We use Vector Space Model to represent a document. Each word (except for the stop words) in the abstract of document will be an attribute of the vector, and the vector space is stored in the Lucene[2] index file. After analyzing all the related documents, data center returns a document list, a hierarchical tree of the documents, and some statistical information according to the application template.

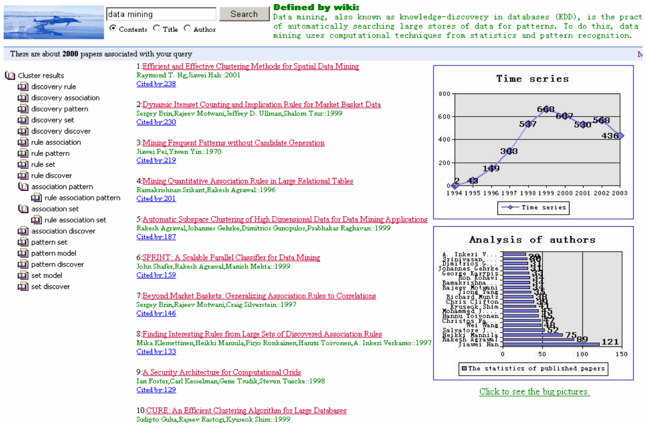
In order to reduce the influence of noisy data, we utilize WordNet[3] to find the wrongly spelled words and the different words which have the same meaning. On the other hand, we get the stop words adaptively according to the statistics on the appearance of the words.

2.5 Query Processor Module

The main task of query processor is to execute query and provide hierarchical clustering results. We partly borrow the thinking from a global frequent item set hierarchical clustering strategy (FIHC[1]) to build clusters. This algorithm can be summarized in three phases: First, initial clusters are constructed. Second, a cluster (topic) tree is built. Finally, the cluster trees are pruned in case there are too many clusters. Furthermore, we refine the FIHC algorithm in the labeling aspect to give user a better understanding of the clustering result.

3. USER INTERFACE

A friendly browser-based user interface is presented to the end users finally. After submitting query keywords, the user will get a comprehensive result shown in Figure 3 which is composed of 5 major parts—the document list (middle column in the figure) ranked by the times of being cited, the clustering result (left column in the figure), definition of the query, time series of the published papers on this topic and the top 20 authors who are the most prominent ones in this field. And by clicking each of the labels of clustering result, users can get some analysis for each sub-topic respectively, the topics are clustered hierarchically. .



data mining
Search
● Contents ○ Title ○ Author

Defined by wiki:
Data mining, also known as knowledge-discovery in databases (KDD), is the pract
of automatically searching large stores of data for patterns. To do this, data
mining uses computational techniques from statistics and pattern recognition.

There are about **2000** papers associated with your query

Cluster results
- discovery rule
- discovery association
- discovery pattern
- discovery set
- discovery discover
- rule association
- rule pattern
- rule set
- rule discover
- association pattern
  - rule association pattern
- association set
  - rule association set
- association discover
- pattern set
- pattern model
- pattern discover
- set model
- set discover

1:Efficient and Effective Clustering Methods for Spatial Data Mining
Raymond T. Ng,Jiawei Hah::2001
Cited by:238

2:Dynamic Itemset Counting and Implication Rules for Market Basket Data
Sergey Brin,Rajeev Motwani,Jeffrey D. Ullman,Shalom Tsur::1999
Cited by:230

3:Mining Frequent Patterns without Candidate Generation
Jiawei Pei,Yiwen Yin::1970
Cited by:219

4:Mining Quantitative Association Rules in Large Relational Tables
Ramakrishnan Srikant,Rakesh Agrawal::1996
Cited by:201

5:Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications
Rakesh Agrawal,Johannes Gehrke,Dimitrios Gunopulos,Prabhakar Raghavan::1999
Cited by:187

6:SPRINT: A Scalable Parallel Classifier for Data Mining
John Shafer,Rakesh Agrawal,Manish Mehta::1999
Cited by:159

7:Beyond Market Baskets: Generalizing Association Rules to Correlations
Sergey Brin,Rajeev Motwani,Craig Silverstein::1997
Cited by:146

8:Finding Interesting Rules from Large Sets of Discovered Association Rules
Mika Klemettinen,Heikki Mannila,Pirjo Ronkainen,Hannu Toivonen,A. Inkeri Verkamo::1997
Cited by:133

9:A Security Architecture for Computational Grids
Ian Foster,Carl Kesselman,Gene Tsudik,Steven Tuecke::1998
Cited by:129

10:CURE: An Efficient Clustering Algorithm for Large Databases
Sudipto Guha,Rajeev Rastogi,Kyuseok Shim::1999

Click to see the big pictures.

Figure 3. User interface of our system

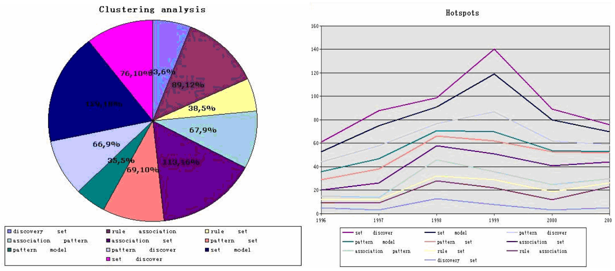


Figure 4. The proportion and time series of each sub-topic

In addition, if a user is interested in a particular author, the system provides the time series analysis of papers published by the author. And the user can also get the answers for the most related coauthors too.

4. CONCLUSION

This paper presents an academic search engine, which is developed as an efficient tool to generate a survey automatically. Moreover, some searching and indexing methods for underlying XML data are exploited. .

5. ACKNOWLEDGEMENT

This work is partial supported by the National Natural Science Foundation of China under Grant No. 60673137 and the National Hi-tech R&D Program of China under Grant No. 2006AA01Z103.

6. REFERENCES

[1] Fung, B., Wang, K., & Ester, M. (2003, May). Hierarchical document clustering using frequent items. SDM'03, San Francisco, CA

[2] http://lucene.apache.org

[3] http://wordnet.princeton.edu

[4] http://scholar.google.com