Research and implementation of search engine based on Lucene

Wan Pu, Wang Lisha

Physics Institute, Zhaotong University,Zhaotong,Yunnan, PR China

Physics Institute

Zhaotong University, Zhaotong,Yunnan, PR China

wan-pu@163.com

Keywords: Search engine; Lucene; web spider; Chinese word segmentation

Abstract. From in-depth research on the basic principles and architecture of search engine, through secondary development for Lucene development package, this paper designs an entire search engine system framework, and realizes its core modules. This system can make up for the deficiency of the existing Lucene framework, and enhance the accuracy of the search engine system, so has higher real and commercial value.

Introduction

With the continuous expansion of network coverage and the development of network technology, the network information resources have been rapidly spread and increased. Large amounts of network information resources from all walks of life, including the information from different disciplines, different areas, different fields, different languages, very rich, and exist with text, images, audio, video, databases and other forms. Internet information has been hundreds of millions, so how to find the needed information from them has become a very important research topic in Internet technology. To help users find the information they need, the search engine came into being.

Search engine is a search tool to help Internet users to query information, it is to collect, find information in the Internet with a certain strategy, and then understand, extract, organize and process the above information, thus providing users with search services, and information navigation purposes achieved. The advent of search engine for our fast, accurate and efficient access to network information resources provide great help. It is a web-based tool developed for the needs of people searching for network information, is the Internet information query navigation, and the bridge between users and network information.

Principle analysis of search engine

The basic principle of search engine is to start from the existing resources, through their summary and link to determine the new information points needed to search for, and then by the relevant program search engine designed traverse these points, finally index, classify, and organize the documents on these points to the index database [1] . Logically this recursive traversal method can put all the information into the index database. When users use a search engine, enter the keywords of the content required to be found, the search program will read the information has been traversed and stored in the index database, to match with the user keyword, and then retrieve the corresponding or related information to output to the user through a certain organization method.

© 2016. The authors - Published by Atlantis Press

A. Search engine system workflow

A search engine to meet users needs is generally consists of information collection, information preprocessing, user interface, as shown in figure 1.

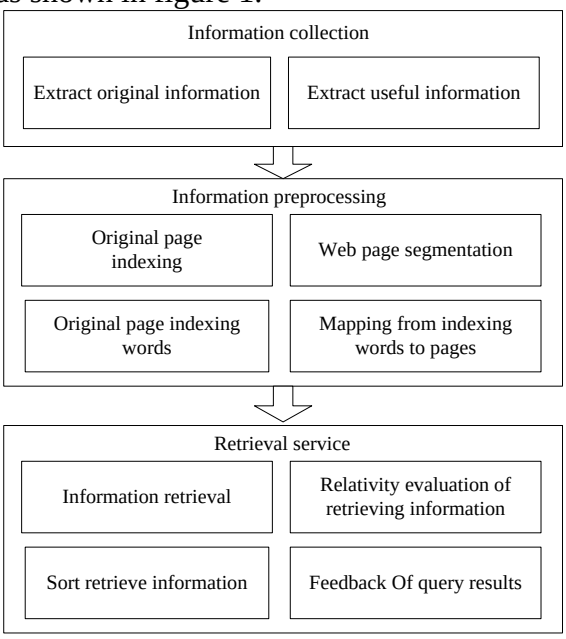

Fig.1 Search engine workflow

Information collection: web page collection obtains input from the URL database, to parse the address of a Web server in URL, establish a connection, send requests and receive data, and then store the obtained Web page data in the original page library, and from which to extract the link information to put into the page structure library, at the same time put the URL to be crawled into URL library, to ensure that the entire process is iterative, until URL library is empty[2]. Information preprocessing: after Web information collection, the preserved page information has been saved in a specific format. So the first step in this part is to index the original page, with that to provide the web page snapshot function for search engines; next Web page segmentation to the index page library, to transform each page into a set of a group of words; finally transform the mapping between Web page and index words into converse, to form inverted file, while gathering the unrepeated index words included in the Web page to be converging vocabulary; in addition, based on the structural information among Web pages to analyze the importance of the information, and then establish page meta-information. Retrieval service: the data delivered to the service stage includes index page library, inverted file and web page meta-information. Query agent to accept user input query phrase, after segmentation, retrieval from the index vocabularies and inverted file to get documents including query phrase, history log and other information, and then calculate the importance of the result set, finally sort and return to the user[3].

Through the above several components, a search engine system can be built, when the user input the keywords, phrases related o the information and resources to be found, the system will traverse the program in accordance with its design search, from the Internet link address traverse pages, the results will be saved to the index database, and then process, integrate the indexed data, finally optimize of the results, according to certain priority algorithm to sort the results, and then store in the index database. When a user types a keyword the search engine will search for the matched page or data information from the index database, and in a certain way show it to the user through the user interface.

B. Key technology of search engine system

A typical search engine structure generally consists of three modules: network spider, indexer, and searcher. Web spider generally first obtains URL from the URL queue to be visited, according to which get the page from Web and analyze it, to extract all of the URL links and add them to the page data URL queue to be accessed, at the same time move the visited URL to the visited URL queue[4]. Continuously repeat the above procedure. All the collected pages to be saved to store for further processing. Initially, in the URL queue only seed URL can be as a starting point the spider traverses the network, generally choose relatively large and very popular website address as a seed URL, because that such pages often have a lot of links to other pages. Web spiders use HTTP protocol to read Web pages and automatically access network resources along an HTML document hyperlink. You can use the network as a directed graph to deal with, each page as a node of it, and the page hyperlink as its directed edge. So you can take a directed graph traversal algorithm to traverse the network.

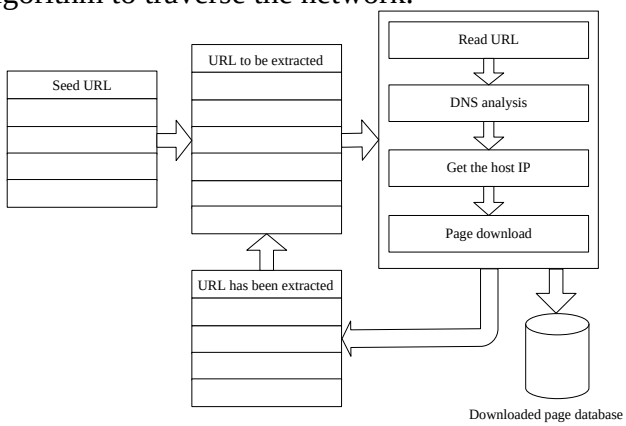

Fig.2 Web spider work principle

Indexer function is to understand the searcher searched information, and extract the index entry, to be used to indicate documents and the index table to generate the documents. Generally index table uses some form of inverted list, that is, finds the corresponding documents from the index entries. Index table also may need to record the position of index entries appear in the document for indexer to calculate the adjacent or close relationship among the index entries. Indexer can use a centralized or distributed indexing algorithm[5]. The effectiveness of a search engine largely depends on the quality of index. When a user query is completed, the search engine has not real-time data retrieval on the web, and the searched data is actually the web data collected in advance. To achieve fast access to the collection page, it must be done through some sort of indexing mechanism. Page data can be represented by a series of keywords, and from the retrieval purposes they describe the content of the page. Just find the page, they can be found. Conversely, if the establishment of the page index is based on keywords, the relevant pages will be quickly retrieved. Specifically, keywords are stored in the index file, for each keyword there is a pointer list, in which each pointer directs to a page related to the keyword, and all pointer lists constitute placing file.

The function of searcher is to quickly detect a document from the index database based on user query, evaluate the association of documents and queries, and then sort the results will be output, finally achieve some sort of user relevance feedback mechanism. The commonly used information retrieval models are set theory model, algebraic model, probabilistic model and hybrid model. Searcher is a module has a direct interaction with the user, and on the interface there are several implementations, the commonly used is Web mode, through these methods, the searcher receives a user query, and carries out

word processing for it, finally obtains query keywords. Based on the above, the Web data matched with the query keyword will be obtained, and returned to the user after sorting[6].

Search engine based on Lucene

Lucene is a full-text retrieval tool package based on Java, it is not a complete search application, but to provide indexing and search capabilities for applications. Currently Lucene is an open source project in the family of Apache Jakarta, also the most popular open full-text retrieval package based on Java, at present there are already many application search function is based on Lucene[7]. Lucene can establish indexing for the data with text type, so you just convert your index data format into text, Lucene will be able to index and search the document. For example, if some HTML documents, PDF documents need to be indexed, they must be first converted into text format, and then given to Lucene for indexing, next, the created index file is saved in disk or memory, finally according to the query criteria entered by the user query the index file. No specifying the format of the document to be indexed also makes Lucene is applicable for almost all of the search applications.

C. Technical analysis of Lucene

Lucene architecture has strong object-oriented features. It first defines a platform-independent index file format, followed designs the core components of the system as abstract class, the concrete platform realization part as the achievement of the abstract class, in addition the platform-related part such as file storage is also packaged as a class, after object-oriented processing, finally a search engine system with low coupling, high efficiency, and easy secondary development is obtained. Lucene system structure is shown in figure 3.

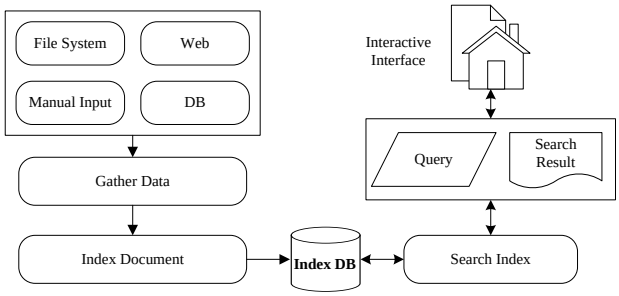

Fig.3 Lucene system structure

In Lucene file format, byte is the basis to define the data types, thus ensuring platform-independent, which is also the main reason for the Lucene index file format and platform independent. Lucene index consists of one or more segments, in which each segment composed by a number of documents. Document object can be treated as a virtual document: for example, a web page, an E-mail message or a text file, then you can retrieve large amounts of data. A Document object contains one or more fields with different domain name, and the field represents this document or some metadata related to it. Each field corresponds to a piece of data, and the data may be queried or retrieved in the index during the search process. The field consists of domain name and value. Term is a basic unit for the search, as field object, it includes a pair of string elements: respectively corresponding to the domain name and value. The conceptual structure of Lucene index files is shown in figure 4.

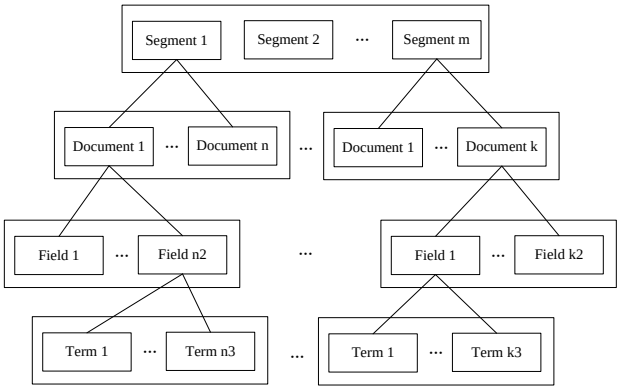

Fig.4 Conceptual structure of Lucene index files

Use of segments can quickly add new documents to the index through adding documents to a newly created index segment and only periodically merging with other existing paragraphs. This process increases the efficiency because that it minimizes the modification of the index file physically stored. One of the advantages of Lucene is to support incremental index. After adding a new document in the index, you can immediately search the contents of the document. Lucene supports for incremental index makes Lucene suit for the work environment of large amounts of information processing, in this environment the method of rebuilding index will look inefficient. Mapping to structure from concept, index is treated as a directory (folder), all the files contained in which are its contents, and these files are stored in group according to the different segments they belonged, the files in the same group have the same file name, different extension names. In addition there are three files, separately used to store the record of all the segments, save the record of deleted files, and control the synchronization of reading and writing, which are segments, deletable and lock files, with no extension names. Each segment contains a set of files, their file extension names are different, but the file names are all the names stored in the file segments.

Lucene system structure has object-oriented feature. Developers do not need to know the internal structure and implementation of Lueene, but simply need to call application interfaces Lucene provided, and they also can extend their own needed functionality according to the actual situation. In the index, Lucene is different from the most search engines, while establishing the index create a new index file, for different update strategies, it combines the new index file with the existing index file, thus greatly improving the efficiency of the index. Lucene also have incremental indexing function, can make batch indexing, and optimize it, the incremental index with small quantities, so for large amounts of data index has obvious advantages. Lucene uses a common data structure to accept the index input, so can be flexibly adapted to a variety of data sources, such as databases, office documents, PDF documents and html documents, etc., when data indexing, only needs an appropriate parser to convert the data source into the corresponding data structure. Although Lucene has powerful search and indexing capabilities, but it is not a complete search engine, cannot collect the information of Internet pages, and in sorting have yet to be perfected[8]. The sorting of search results is very important for the search engine, usually users only take attention to the first page search engine returned, therefore, taking the pages valuable for users, with high level as the top surface of the page is an important topic of search engine study.

D. Search engine based on Lucene

Search engine mainly consists of collecting, indexing, and retrieval system, while the user interface is a way to display search results for users. Web spider in the network according to a certain strategy to extract pages and recursively download the crawled pages. Indexing system for the pages the web spider have collected uses analysis system for word segmentations, then get the corresponding index entry, and

for all types of documents, uses the corresponding parser to parse the text, then index file and store it in the index database. Users input the search keyword through the user interface, and then the retrieval system will analyze it and submit it to the word segmentation system for processing, match the keywords obtained from the above processing with the words have been indexed, by specific algorithms sort the pages with same or similar keywords, finally return the search results to the user interface. Search engine overall structure shown in figure 5.

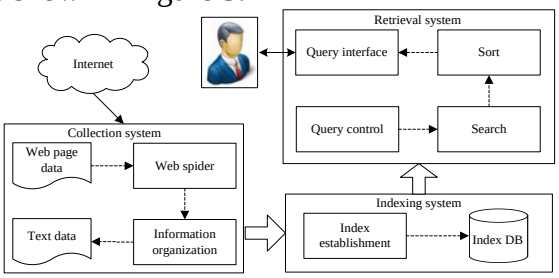

Fig.5 Overall structure of search engine

The indexing mechanism in Lucene system should have analytical function, Lucene itself has the function to analyze txt, html files, and because of many Internet file formats, so in order to achieve a variety of document analysis, the corresponding search package needs to be added. Lucene analyzer consists of two parts: one part is the word segmentation device, being called Tokenizer; the other part is a filter, known as TokenFilter. A parser often consists of a word segmentation device and a plurality of filter, in which the filter is mainly used to deal with the segmented words. In the index establishment, what can be written in the index and retrieved by users are the entries. In fact, the so-called entry is the text after analyzer word segmentation and related processing. Word segmentation device through a next () method itself provided returns a primitive, segmented entry, and the filter through this method returns a filtered entry, with no segmentation function. As the filter constructor receives an instance of TokenStream, there will be two situations: first, the filter and other filters can be nested together to form a nested pipe filter structure; second, the filter can be combined with tokenizer to filter the segmented words from it. This nesting forms the core structure of Lucene analyzer.

Retrieval function is the last link to achieve search engine, and the important factor to measure it in response speed and result sort. When a user enters a search keyword, the word segmentation system to analyze and cut, then the similarity calculation and matching with the morpheme vector in index database, and finally the search results successfully matched will be returned to the user. Retrieving part of the search engine system consists of Lucene search statement analysis system and search result clustering analysis system, in which the former is to understand the user input keywords, according to the reverse maximum matching algorithm for retrieve word segmentation, if the segmented results need to pause word filter, it needs to deal with the ambiguity field by using word segmentation probability, and then get the actual semantic words, establish a search term. Then, the Lucene search system query and submit the results to the clustering analysis system to analyze and process, so find high correlation pages and automatically generate pages. Finally, the analysis system will detect the similar documents from the Lucene search results.

Conclusion

The rapid development of the Internet, the amount of information is increasing exponentially, but the ultimate goal is to enable users to easily access the information, this mission falls on the search engine, furthermore, how to return the needed information, web content with high quality to users, presents higher requirements and challenges to the search engine. Because that the Lucene scoring algorithm has

not well reflected the page location information in the website, this paper designed an improved solution in index and retrieval module, which can well unite the basic points of the document, and the document location information in the website, as well as the document characteristics, to improve the accuracy of search result sorting, thereby enhancing the accuracy of the search.

References

[1] Monz C. Proceedings of 25th European Conference on Information Retrieval Research [c], Berlin/Heidelberg: Springer, 2003:571-579.

[2] Nicholas Lester, Justin Zobel, Hugh E Williams. Efficient Online Index Maintenance for Contiguous Inverted Lists [J], Inf. Process. Manage, 2006, 42 (4): 916-933.

[3] George Samaras, Odysseas Papapetrou. Distributed Location Aware Web Crawling, In Proceedings of the 13th international World Wide Web conference [J], New York, USA：ACM Press, 2004: 468-469.

[4] Hai Zhao, Changning Huang. Effective tag set selection in Chinese word Segmentation via conditional random field modeling [C], In: Proceedings of PA-CL IC220.WuHan, November 123, 2006: 84-94.

[5] Arvind Arasu, Jasmine Novak, Andrew Tomkins, John Tomlin. Page Rank Computation and the Structure of the WEB: Experiments and Algorithms, In Proceedings or 11th International World Wide Web Conference, 2002.

[6] Giuseppe Antonio Di Lucca, Anna Rita Fasolino, Porfirio Tramontana. Reverse engineering web applications: the WARE Approach [J], Journal of Software Maintenance and Evolution: Research and Practice, 2004, 11(3):15.

[7] Giuseppe Pirro, Pomenico Talia. An approach to Ontology Mapping Based on the Lucene Search Engine Library[C], Proceedings of the 18th International Conference on Database and Expert Systems Applications, 2007, 9:156-158.

[8] Laurence Hirsch, Robin Hirsch, Masoud Saeedi. Evolving Lucene Search Queries for Text Classification[C], Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, 2007, 6(12):166.