International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-4 Issue-6 December-2012

Study of an Intelligent Spider Algorithm & Metasearch Engines

Govinda Borse[1], Ankita Baheti[2]
SS, Technocrats Institute of Technology Bhopal, India [1]
CTA, Technocrats Institute of Technology Bhopal, India [2]

Abstract

The main aim of search engines is to provide most relevant documents to the users in minimum possible time. People search for a variety of reasons. A big reason to search is to look for something known to exist. Determining a user's preference of Web searches is a difficult problem due to the large amount of data available concerning the searcher. Indexing is performed on the web pages after they have been gathered into a repository by the crawler. In this paper we have made the survey of working of search engines & also analyzed the highly scalable & potential metasearch engines. A metasearch engine is a system that supports unified access to multiple local search engines. It queries other search engines and then combines the results that are received from all. In effect, the user is not using just one search engine but a combination of many search engines at once to optimize Web searching.

Keywords

Crawlers algorithm, Intelligent Spider, Metasearch Engine , Indexing

1. Introduction

The World Wide Web contains hundreds of thousands of electronic collections that often contain high quality information. The main purpose is to select the best collection of information for a particular information need. Often, a user knows something exists and just needs to find it within a site. Search functions appeal to power users, frequent visitors, and the plain impatient that are all looking to find a result quickly. A well-executed search facility is one major advantage a Web site has over printed media, as it gives users greater control over a site's content, allowing them to filter it to just what they want to see. Searching desired data is one of the most famous ways the Web is utilized. Many search engines have been created to facilitate the retrieval of web pages. Search engines mainly help the users to find the desired pages & retrieve all related pages to the users. So how do search engines work?

2. How Search Engine Works?

First, a large number of pages are gathered off a Web site using a process often called *spidering*. The algorithm used for the spidering process is also named computer robot or spider, wanderer, web crawler. The spider acts as information retriever for some web service systems and search engines. Next, the collected pages are indexed to determine what they are about. Finally, a search page is built where users can enter queries in and get results related to their queries.

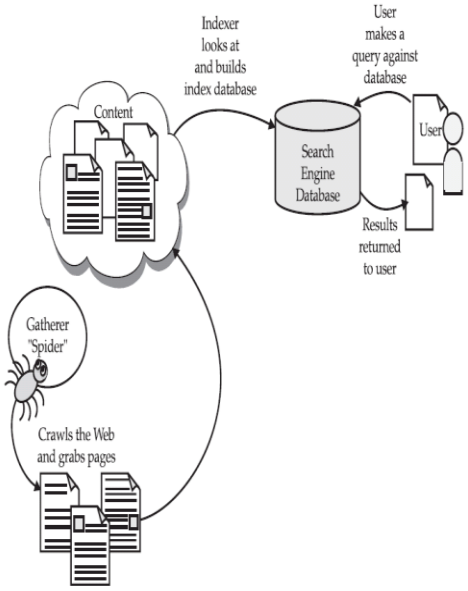

Fig.1: Overview of working of search engines

- **Gathering Pages**

Search engines use spider to gather pages of the Web for indexing. Spiders start their gathering process with a certain number of starting point URLs and work from there by following links. When the spider visits the various addresses in the list, it saves the pages or portions of the pages for analysis and looks for links to follow. For example, if a spider were visiting the URL http://www.google.com, it might see links emanating from this page and then decide to follow them. Not all search engines necessarily index pages deeply into a site, but most tend to follow links—particularly from pages that are well linked

themselves or contain a great deal of content.

Indexing Pages

After Gathering pages, the next step is the indexing pages. This step is attempting to determine what a page is about. This is usually called *indexing*. The method each search engine uses varies, but basically an indexer looks at various components of a page, including possibly its **<title>**, the contents of its **<meta>** tags, comment text, link titles, text in headings, and body text. From this information it will try to distill the meaning of the page. Each aspect of a page might have different relevance, and within the actual text, the position or frequency of different words will be taken into account as well. However, not all content within a page matters to a search engine. For example, *stop words* are words that a search engine ignores, normally because they are assumed to be so common as to carry little useful information. Examples of stop words might be "the," "a," "an," and so on. Most search engines have some stop words, but some engines like AltaVista claim to even index common stop words like "the." While the use of stop words may improve a search engine by limiting the size of the index file and focusing it on more content words, it may not match how users think about queries. Once a page has been analyzed for the various keywords, it is ranked in relation to other pages with similar keywords and stored in a database. Ranking is the very secret part of search engine operation. How a particular search engine decides one page should be ranked higher than another is what search engine promotion specialists are always trying to figure out. A very popular way to rank pages today is based upon determined site landmarks. Home pages and major section pages may be given higher weight than other pages in a site. Pages that have numerous incoming links will also be given extremely high ranking.

3. Relation Web Pages

The relation between different web pages is based on the Web page linkage structure. This relation can be defined as a directed graph G= (W, V). Here W is representing the collection of web pages in internet, & V is a set of url_{ij} that the linkage exists between web page i and web page j. For searching the web page in internet an information retriever is important. The linkage structure among web pages decides the implement method of information retriever. There are two different assumptions on which the linkage structure has based:-

1. A hyperlink given from the page 1 to web page 2 is recommendation of page 2 by the author of page 1.
2. If web page 1 & web page 2 are connected by a hyperlink, then they might have the same topic.

There are some main steps which are basically followed according to the principle of spider algorithm. These steps are:-

1. URL table of spider is firstly initialized in the first step.
2. Gets the order of the web page obtained by page rank of URL.
3. The content of web page including user query requirement is loaded.
4. Parses new URL in web pages retrieved from Internet, and adds the new URL and it's keywords into some index database
5. Page rank is updated by computing the page rank, addition of some URL which satisfy user query requirement to URL table of spider, then go to step 2, until not new URL is found.

4. The Spider-An Intelligent Algorithm

During the starting or the commencement, when the user uses the spider, it is a weak program. But the intelligence of this algorithm is improved when the user uses spider time & times. The spider basically can collect the actions performed by the various web users. It can also save the knowledge to a database which is also called as web user log & so we can easily use these history knowledge stored in web user log. Because of this spider becomes wiser & wiser. The specific structure of web user log is:- (User-id, Date, Start-time, End-time, Key-words, Interest-rank).

- **User Search Space**

When the user enters some query keywords to the search engine, the search result directly is selected from an index database in which every record have stored the URL and keywords of web page of URL in web page server. The URL and web page have retrieved by spider of search engine. And so, in these styles, the search space of the user query keywords is apparently made up of all web pages in internet. But we regard that the spider should be materialized some factors, in order to improve the search precision which the currently spider spreads all internet. In order to improve the search speed, the currently search engine matches completely user query keyword and the search result. But it must be changed these approaches to spread and match, the main reasons see below:

1. The user query keywords are relational to the user history knowledge; the spider should make full use of the user's history knowledge of web user log in process of crawling web page.
2. The search result should satisfy user query requirement. For the user, it is impossible to need amount of web page in short time, the user want to find some fewness, almightiness, exactness, real-time information, where fewness means that the limited and usefully information is return; almightiness means that the information relating to user query keywords will be return; exactness means that the user will get all information of his query requiring information; real-time means that the user should get some new information and wait result in short time.
3. The user query keywords reflect a special meaning and special information space, the space should be constructed by a few part of web pages in internet. These web pages are relational to user query keywords.

By collecting all the reasons, we thought the web pages, which constructed the search space of the user keywords, forms a forest (as in figure), it makes up of some interest tree in which every nodes appends a special mark note-URL. A spider can only crawl in fields of web pages in the tree. How to generate the interest tree? The spider starts to work from an initiated URL table, every URL in an initiated URL table is root node of the interest tree, the other nodes of the interest is these web page crawled by the spider in accord to the strategy.

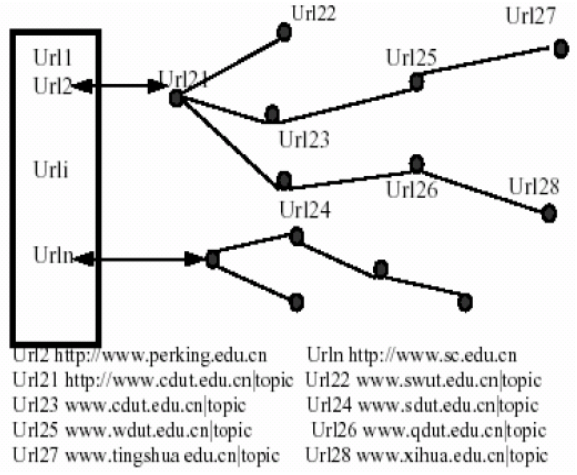

Fig. 2: User search space

5. The Crawling Algorithm

The spider algorithm crawls around the web to find out the related searching keyword or the document, which the user wants to search. So it is also called as Crawling Algorithm. When this algorithm starts to work, users select some URL, from user web log database. Thereafter, we construct an interest tree T_{ri} from URL_i. The number of URLs which we select URL from the user web log database decides the number of the tree in G'. In order to improve the crawling speed of the spider, we adopt the agent program or the multiple-thread program. Every agent or thread program competes to crawl the web page of an interest tree. A key technology of the intelligent spider is how it constructs the user interest tree. The working principle of intelligent spider is divided into different steps. The figure shown below shows the illustration.

1. GetDocument (url:string) snatches web page in internet and parses some useful URL, and return a text, some URL his is simply to download the template, and replace the content with your own material.
2. MatchAnchor(URLS) matches the user query keywords with anchor text of the parameter URLs and pushes the URLs, which user query keywords match successful with anchor text of parameter URLS, into the waiting stack, if user query keywords does not match successful with anchor text of parameter URLS then spider call MatchTopic(URLS). It can match the user query keywords with the title text of parameter URLS, if the user query keywords matches successful with the title text of parameter URLS, then the spider program pushes these URL into the waiting stack, if it failed, then the spider calls MatchContent (URLS). MatchContent (URLS) returns the URL which user query keyword matches successful with title text of parameter URLS, if the web page content of parameter URLS matches successful with the user keyword, then the spider push these URL into the waiting stack and save the keywords of web page to database which spider return to the user, else stop spider to continue.

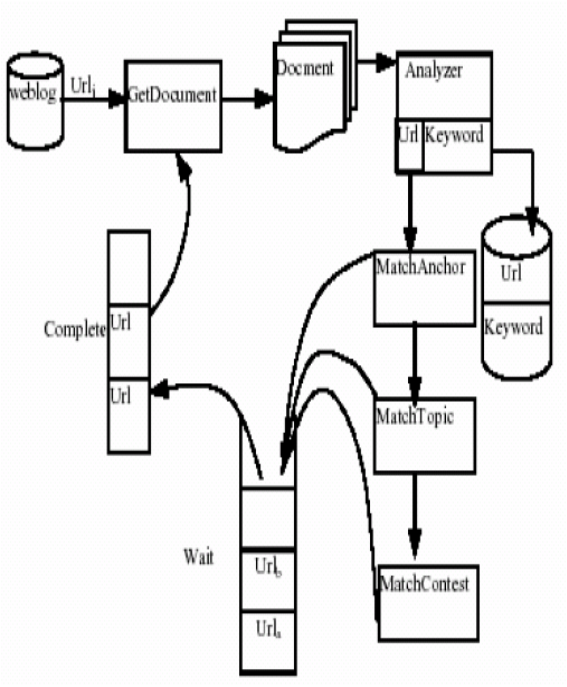

Fig.3: Illustration of working principle of Spider Algorithm

3. Analyzer (f: text) analyzes web page document. This function picks up some URL and syncopates words, finds some main keywords by the word frequency statistics method and saves (URL, keyword) to database which spider return to user.
4. The completed stack will save some URLs, which the spider have crawled URLs of web pages. Some URLs which save the waiting stack can't save the completed stack. If the URLs have saved the completed stack, the spider gives up these URLs and takes out her URLs from the waiting stack.
5. It is availability to crawl from one of Initializes URL table and spread all over the user interest tree, and so the waiting stack and completed stack is designed.

6. An Effective Metasearch Engine

There are many search engines on the Web, which are attempting to index the entire Web and provide a search capability for all web pages. But these centralized search engines suffer from a number of limitations. The limitations are:-

1. The coverage of the Web by each of them is limited due to the lack of appropriate links or any other reason.
2. As the major search engines get larger, higher percentages of their indexed information are being obsolete.

There are many search engines that focus on documents in confined domains such as documents in an organization or of a specific subject area. For example- The search engine scirus (http://www.scirus.com) mainly focus on scientific research tool available on the web. Many organizations have their own search engines. There is reason to believe that all these special-purpose search engines combined together can provide a better coverage of the Web than a few major search engines combined. Thus, an alternative approach for providing the search capability for the entire Web is to combine all these special-purpose search engines. This is the metasearch engine approach. A metasearch engine is a system that supports unified access to multiple local search engines. It does not maintain its own index on web pages but a sophisticated metasearch engine often maintains characteristic information about each underlying local search engine in order to provide better service. When a metasearch engine receives a user query, it first passes the query (with necessary reformatting) to the appropriate local search engines, and then collects (sometimes, reorganizes) the results from its local search engines. In addition to the potential of increased search coverage of the Web, another advantage of such a metasearch engine over a general-purpose search engine is that it is easier to keep index data up to date as each local search engine covers only a small portion of the Web. In addition, running a metasearch engine requires much smaller investment in hardware (computers, storage devices) in comparison to running a large general search engine such as Google which uses thousands of computers.

7. Conclusion and Future Scope

Here our study defines the working of an Intelligent Spider algorithm. This study shows that Web search engines continue to have different abilities and the overlap among Web search engine results. Web meta-search engines each provide a different and unique perspective on the Web.

The spider is the general model, in the future, we let spider learn the mankind intelligence to search information. There are five intelligence behaviors about human being: fell behavior, memory behavior, study behavior, thought behavior, comprehension behaviour.

Nutch is an open source search engine, builds on Lucene and Solr. According to Tom White, Nutch basically consists of two parts: crawler and searcher. The crawler fetches pages from the web and creates an inverted index from it. The searcher accepts user's queries to the fetched pages. Nutch can run on a single computer, but also can works great on multinode cluster. Nutch use Hadoop MapReduce in order to work well on distributed environment. Hence we built simple crawling technique using currently developing search engine Nutch.

References

[1] Yajun Du, Haiming Li, Zheng Pei, and Hong Peng, "Intelligent Spider's Algorithm of Search Engine Based on Keyword," ECTI transactions on Computer and Information Theory VOL.1, NO.1, MAY 2005.

[2] Zonghuan Wu, Weiyi Meng, Clement Yu, Zhuogang Li: "Towards a Highly Scalable & Effective Metasearch Engine".

[3] Parul Gupta, Dr. A.K. Sharma," Context based Indexing in Search Engines using Ontology," 2010 International Journal of Computer Applications (0975-887) VOL. 1- No. 14.

[4] Fabrizio Silvestri, Raffaele Perego and Salvatore Assigning Document Identifiers to Enhance Compressibility of Web Search Engines Indexes In the proceedings of SAC, 2004.

[5] Bernard J. Jansen, Danielle L. Booth, Amanda Spink," Determining the User Intent of Web Search Engine Queries".

[6] Jansen, B. J. and Spink, A. 2005. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing & Management. 42, 1, 248-263.

Govinda Borse completed the BE degree in IT from North Maharashtra University in 2010. He is currently pursuing PG under from Rajiv Gandhi Technological University.

Ankita Baheti completed the BE degree in Computer Engineering from Mumbai University in 2010. She is currently pursuing PG under from Rajiv Gandhi Technological University.