D’Galaxy: An Information Retrieval System for Intranet Search

Su Cheng Haw¹, Yeong Wai Loh², Saik Keong Lua³, Lay Wen Ng⁴, G.S.V. Radha Krishna Rao⁵

Faculty of Information Technology
Multimedia University
63100 Cyberjaya, Malaysia
¹,⁵{schaw, gsvradha}@mmu.edu.my, ²yeongwai@hotmail.com, ³stanleylua@yahoo.co.uk, ⁴nglaywen@yahoo.com

Abstract

There are enormous amounts of information widely available in the Intranets. This information is only useful if data can be retrieved in an accurate and timely manner. Currently, Intranet search engine has become a necessity due to lack of an efficient way to disseminate useful information to its members. In this paper, we examine the importance of Intranet search engines and propose D’Galaxy as the search engine in Multimedia University (MMU). This intranet search engine will be of content-based using the full-text indexing.

Keywords: intranet search, information retrieval, web crawler, full-text indexing

1. Introduction

Before the emergence of Internet search engine, people were confined to traversing through sites they had already known of, in a hopes to find a useful link. This may have been adequate in the early days of the Internet, but as the Internet continued to grow exponentially, it became necessary to develop a means of locating desired content [1]. Search services are now among the most frequently visited sites on the Web with millions of hits every day [2].

The term “search engine” is often used generically to describe both crawler-based search engines and human-powered directories. These two types of search engines gather their listings in radically different ways. Crawler-based search engines, such as Google, create their listings automatically. They “crawl” or “spider” the web. If your web pages are modified, crawler-based search engines find these changes and regenerate the listing. Page titles, body copy and other elements all play a role.

Human-powered directories such as the Open Directory, depends on humans for its listings. A short description is submitted to the directory or editors will write one for sites they review. The search looks for matches only in the descriptions submitted. In this case, if there are changes to any of the web pages, it has no effect on the listing. The only exception is that a good site, with good content, might be more probable to get reviewed.

Intranet is network of computers that can be accessed only by an authorized set of users within an organization. Its purpose is typically to share information and computing resources among employees within an organization. Hence, an intranet search engine is designed to crawl and index internal web servers or portions of these servers to create custom, searchable indexes.
of the documents housed on the servers. Intranet search engines have similar features with their Internet counterpart but can be designed with some unique features.

There are two types of Intranet search, namely desktop-based and web-based [3]. Desktop-based address the whole spectra of electronic information that might be found in an organization, including video, images, database, spread sheets, email, semi-structured data and so on. In an Intranet the users are no longer considered anonymous, as an intranet is a closed network such as employees, managers, visitors, customization can be used to enhance information retrieval. A web-based search engine focuses on HTTP-accessible documents only. However, our research has focused on mainly the web-based search engine utilizing full-text indexing.

Without a search engine, no matter how well sites are designed and organized, finding information would be much tedious and time consuming [4]. In this paper, we propose the architecture and algorithms for D’Galaxy, an Intranet search engine. Our search engine is capable to performs the following functions:

- D’Galaxy is created for organizing intranet search within our university, MMU network. By restricting the search space, the system can be used to help users find only related Intranet sites with higher accuracy instead of wandering through the Internet.
- D’Galaxy will return the ranking results based on their relevancy to the user. This is possible by using full-text indexing technique with Boolean capability. This reduces the searching time that best matches with the user needs.
- This Intranet search engine checks the search logs every time the page is submitted. This reminds user to enter a search criteria before the page is submitted. A couple of other tips as a bulleted list are available to help users.
- D’Galaxy display a report of the top search queries, total result found and time needed for the search.
- The system also implements a spell checker to check the spelling of the query. If user enters words that are not semantically correct, spell checker will determine the nearest matching keyword.
- User interests may change over time. To reflect the true state of an individual’s preferences, user can select the total results per page to be displayed based on their personalization. Besides, user can choose the background color of the results page.
- The system is equipped with a dictionary function in which user can click on the word to check for their meaning. The query will be broken down into individual entities, so that the user can click on each entity to refer their meaning.
- The system also provides a simple description of the Boolean operator supported. This feature can minimize the user searching time in which user can apply the Boolean operator to find results that best suite their keyword.

There exist several types of searching technologies. One of them is the Meta tag search. This technique is convenient but is far from perfect as it relies solely on the web authors own responsibility to describe the contents of his website. Web authors can easily fool search engines ranking by flooding their tag with frequently searched terms. To avoid this matter, we plan to avoid relying on Meta tags but to use the content of the webpage itself as reference via the full text indexing technique.

The outline and main contributions of the paper are as follows: Section 2 discusses related work. Section 3 presents the architecture diagram of Intranet search
D’Galaxy: An Information Retrieval System for Intranet Search

2. Related Work

2.1 Web evolution

Alan Emtage, a McGill University student, created Archie, one of the first attempts at organizing information on the Internet in 1990. Archie archived the repository of Internet files, Anonymous File Transfer Protocol (FTP) sites. By December 1993, a new robot now known as spiders came into the searching scene. Repository-Based Software Engineering (RBSE) made its important debut by listing the results based on relevancy to the keyword. Prior to that, the results were unordered and finding the right location could require traversing through hundreds of listings. In February 1993, Stanford students launched Excite. It introduced concept-based searching. This engine utilized statistical word relationships, such as synonyms and also suggests some possible action to be taken if the exact keyword was not entered correctly. Brian Pinkerton of the University of Washington developed WebCrawler in April 20, 1994 [5]. WebCrawler indexes the entire text of web pages while the other search engines only indexed the URL and titles.

The first browsable search directory was EiNet, now known as Tradeway Galaxy, which went online January 1994. It made good use of categories and subcategories. This feature enables users to narrow their searching scope. Yahoo perfected the search directory. Yahoo grew out of two Stanford University students, David Filo’s and Jerry Yang’s as a way to keep track of their personal interests in April 1994 [5]. Yahoo’s user-friendly interface and easy to understand directories have made it the most used search directory.

Soon it was the rise of meta-engines. They simultaneously compile search results from various different search engines, and then list the results according to the collective relevancy. The first meta-engine was Meta Crawler released in 1995.

In December 1995, AltaVista was the first to implement natural language queries as well as Boolean search techniques. It was the first to offer “Tips” for good searching prominently on the site. Paul Gauthier and Eric Brewer at Berkeley introduced HotBot in May 1996, which claims to support indexing the entire Web. Then came the cult status Google which started as a research project at Stanford University since late 1997. Likewise, it attempted to improve relevancy rankings. Google uses PageRank, which basically monitors how many sites link to a given page [6]. The more sites that link to a given site would yield higher ranking in the result list.

In 1998, there were two types of search engines: author controlled services such as AltaVista and Excite, in which the results are ranked by keyword relevancy and editor-controlled such as Yahoo and LookSmart in which people manually decided on placement and relevancy.

2.2 Searching and optimization techniques

Web crawlers are central part of search engine. There are many web crawler developed and implemented such as RBSE [7], the WebCrawler [8], the World Wide Web Worm [9], the crawler of the Internet Archive [10], the personal search agent SPHINK [11], an early version of the Google crawler [6], the WebFountain incremental crawler [12], and the WIRE crawler [13].

A number of Web query languages such as W3QS, WebSQL and WebLog have also being proposed to support semistructured data [14]. They aimed to combine the content-based queries in information retrieval with structure-based
queries in DBMS. These approaches consist of a wrapper, which maps a data source to a common data model. However, constructing wrappers is time-consuming and hardly can cater for larger data sources.

Gordon [15, 16] proposed a genetic algorithm-based approach for document indexing. In their proposition, keywords represent a gene, a document’s list of keywords represents chromosomes, and a collection of relevant documents judged by a user represents the population. The population then evolves through generations and eventually finds a set of keywords which, in terms of fitness function, best describes the documents. Chen et al proposed a genetic algorithm-based page clipping synthesis (PCS) [17]. These page clipping provide users the information they are most interested in and therefore save the users time and trouble in browsing lots of hyperlinks. Nick and Themis [18] proposed a genetic algorithm in which the system recommends web pages directly to user based on the user interest information.

Automatic indexing algorithms have been used widely to extract key concepts from textual data [19]. Besides, linguistics approaches such as noun phrasing also have been applied to perform indexing for phrases rather than just words [20]. These techniques are useful especially in extracting meaningful terms from text documents.

Studies had shown that nature and extent of search engine that include personalization features performs better [21, 22]. Currently web sites rely heavily on user inputs for a personalization solution. Mobasher et al proposed a general architecture for automatic Web personalization [23]. This method tracks the user’s interest from the Web server logs. Perkowitz and Etzioni suggested anticipating the user’s need based on his past navigation history [24].

One of the most important determinants of the success of Intranet sites is their degree of usability [25]. Color and navigation is the key aspects that affect usability. Analysis showed that blue links on a white background resulting in better outcomes than black links on a white background. Links may be noticed more accurately and faster if the contrast between link color and background color is maximized [26]. Appearance of screen elements also is a factor contribute to aesthetic preferences of users [27].

3. Overview of D’Galaxy Search Engine

Figure 1 illustrates the architecture of the search engine. The client specifies the query to the search engine. Search engine looks in the index file in database for matches. After search engine gathers matching pages, it will sorts by relevancy and sends the result page back to the client.

Webmaster will indicate the web spider to crawls the web pages. Web crawler is a module aggregating data from the Internet in order to make them searchable. The indexer takes the collection of documents or data and builds list of words and notes where they were found and stores the indexed file into the database.
Systems usually run the crawler and indexer sequentially in cycles. First the crawler retrieves the content, and then the indexer generates the searchable index. Based on the index, the page matching to the criteria specified will be return.

3.1 The Indexer

Full-text indexing literally creates a virtual copy of the entire website. The option is still feasible as it only encompasses Intranet searches. With this, content can be subjected to further scrutiny and hopefully more precise information. The first step is to initiate the creation of an index; this index will contain location information for each and every word in all of your documents. The creation of this index is external of the files and does not affect them in anyway. Indexed documents are typically specified according to directory and extension. There can either be one index for all of the files, or several separate indexes, each for a different project. The indexes automatically are updated when new documents are created, or existing documents are changed. However, any changes to the table’s structure such as configuration data will need a complete rebuilding or the full-text index.

Once there is an index, it can be used to locate, view and retrieve information. Using the indexes created, the search query can be used to locate the required information in your documents. Results are displayed almost instantly, despite its relatively large size and thus proving the speed and advantages of implementing indexes.

3.2 The Crawler

The general algorithm involves backtracking to the root directory and penetrating new web pages via their links. The process continues until the entire website (Intranet) is indexed. Besides, our crawler is able to recognize duplicate pages and discard them accordingly. First and foremost, our web crawler will be given a starting URL as its input and repeatedly executes the following steps as shown in Figure 2(a) and 2(b):

- Take a URL from the list of URLs
- Determine the Internet Protocol (IP) address of its host name to ensure it belongs to the MMU domain
- Extract any links contained within the page.
- For each of the extracted links, ensure that it is an absolute URL and add it to the list of URLs to download, provided it has not been encountered before.
- If it is a relative URL, resolve it by adding it to the base URL.
- Then, process the downloaded document in other ways (e.g., index its content).
- Then begin the process of indexing the content by using the first links from URL lists
- Strip out SCRIPT and STYLE tag, and remove all unnecessary symbols
- Full-text index their content into another table along with their URL.

Then continue recursively extract next URL to index its content.

3.3 Web Spider

To find information on the hundreds of millions of Web pages that exist, a search engine employs special software robots, called spiders, to build lists of the words found on Web sites. When a spider is building its lists, the process is called Web crawling. Web crawling is a process to gather content descriptors from the document collection. In case of html files it follows links to other pages within the site. This is called a site being “spidered” or “crawled”. In the case of remote indexing, the crawler returns to the site on a regular basis, such as every month or two, to look for changes and updates.
Figure 2: The flow of Web Crawler

Once the spiders have finished the task of finding information on web pages, the search engine must store the information in a way that makes it useful. Everything the crawler finds goes into the second part of a search engine, the index. The index or the catalog contains all the descriptors that the crawler finds. If a document collection changes, then this catalogue is also updated with new information. There are two key components involved in making the gathered data accessible to users:

- The information stored with the data
- The method by which the information is indexed

In the simplest case, a search engine could just store the word and the URL where it was found. In reality, this would make for an engine of limited use, since there would be no way of telling whether the word was used in an important or a trivial way on the page, whether the word was used once or many times or whether the page contained links to other pages containing the word.
4. D’Galaxy User Interface

In the main screen, user can enter query in the text-box, selected criteria to narrow down their searching scope and click on the “Submit Search” button as shown in Figure 3.

When user enters a keyword that returns zero results, a spell checker will check the spelling of the query to get the nearest matching keyword and a couple of other tips as bulleted lists are display to help user as shown in Figure 5.

Figure 3: D’Galaxy main screen

Figure 4 shows the results page for the keyword “multimedia”. This page also displays the total results found, execution time and the relevancy of matches found in the database. Besides that, a simple description of each URL is summarized.

Figure 4: Results return for “multimedia” keyword search

5. Algorithm

5.1 URL Crawler

The crawler will crawl through all the pages in the MMU website and branch out accordingly. As it passes through web pages, it will process and return all subsequent links and store in the database as shown in Figure 6.
5.2 Content Crawler

Figure 7 shows how web crawler index the information and content in web page to store in database.

```
CLASS crawlContent
    con = NULL
    Loading MySQL- Java Driver
    stmt1 = SQL statement to select all from URL table
    ur = get result set from stmt1
    url id = page id from ur
    stmt2 = SQL statement to select page id from CONTENT table in a descending order
    url id = get result set from stmt2
    urID = page id from url id
    FOR every page id in the URL table LOOP
        stmt = SQL statement to select page url from URL table
        urlset = get result set from stmt
        url = page url from urlset
        IF url not end with "jpg" AND "mpg" AND "avi"
        AND "wmv" AND "asx" THEN
            IF check URL is equal to TRUE THEN
                CONTENT = get text from text area
                Replace ' ' with *, replace > with empty space,
                remove white space, replace non breaking space
                with empty space in content
                stmt3 = SQL statement to insert page id, page url, page title, page content into CONTENT table
                IF check URL is equal to TRUE THEN
                    CONTENT = get text from text area
                    Replace ' ' with *, replace > with empty space,
                    remove white space, replace non breaking space
                    with empty space in content
                    stmt3 = SQL statement to insert page id, page url, page title, page content into CONTENT table
                    CONTENT = get text from text area
                    Replace ' ' with *, replace > with empty space,
                    remove white space, replace non breaking space
                    with empty space in content
                END IF
                END IF
            END IF
        END FOR LOOP
END CLASS
```
6. Conclusions and Future Work

Running an Intranet search engine is different from merely using a public search service in Internet. This could provide users with information they are most interested in and therefore save the users time and trouble in browsing lots of hyperlinks. D’Galaxy employs a number of techniques to improve the search quality including relevancy ranking, personalization features, supporting Boolean operator, proximity information, spelling checker and thesaurus to aid the user.

6.1 Future work

(a) Enable better query understanding

Although we have implemented a spell checker to aid users in finding the correct word and to solve typo errors, search engines till today still lack the intelligence to actually understand the semantics rather than the syntax of a search query. Currently, semantic searching is still under development phase whereby natural language processing algorithm would be implemented to each query. Query like “Malaysian food” would then yield results like “nasi lemak”, “kuey teow” and etc., food items of Malaysia only.

(b) A better ranking algorithm

D’Galaxy ranks based on the number of occurrence of words in the content and title. Thus the results are accurate base on content. However, our research has proven that this alone is insufficient when the content searched is not purely documented based, as in the case of Internet. GOOGLE currently uses a massive link network (518 million hyperlinks) to help prioritize the results of web keyword searches code named PageRank [6]. A website with many other websites linking to it would mean the website is popular and most likely have high quality content. GOOGLE uses a theme based system where each website is assigned a particular theme. For instance, MMU would have an educational theme and hence would not be mistaken for maybe Managed Municipal United, also MMU.

(c) Multimedia Search Engine

The current version of our Intranet search engine is only capable for searching documents in HTML format. This version could be enhanced by supporting searches for various types of files including images, audio, video, semistructured data such as XML (eXtensible Markup Language) and so on [28]. Moreover, audio based search engine (hands-free operation) could be included to aid visually or physically impaired users, a novel idea that can potentially be a new vanguard in accessibility sector.

7. References


