JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Unveiling WebPageDigger: Empowering Web Page Mining through Internet Traffic Analysis

Mr. DARA MURALI KRISHNA [1],
[1]Senior Lecturer, Department of Computer Engineering
Government Polytechnic, Srikakulam,Andhra Pradesh -532005.

Abstract

Web search engines are essential tools that automatically navigate the enormous internet by downloading files and clicking on links to visit numerous web pages. These engines are essential to information indexing because they help consumers locate pertinent content quickly. Web searches have far more potential than just indexing; they may be used for applications like page validation, structural analysis, visualization, update notification, mirroring, and personal web assistants/agents. Worms, robots, and spiders are other names for web searches. Single query is contained in one machine. The Search just sends HTTP requests for documents to other computers on the Internet, exactly way a web browser does when a user clicks on links. Actually, the Search only ever follows links automatically. In addition to the speed of their own Internet connections, the speed of the websites being searched has an impact on how rapidly someone can search the web. The amount of time spent searching could be significantly reduced if several downloads are made at once, particularly if one is a searching site from multiple servers. In this work, the "Breadth First Searching" algorithm—a modified version of one of the original dynamic Web search algorithms—is used.

Keywords

Internet Connection, Breadth First Search Algorithm, Worms, Robots, Web Search.

1. Introduction

There is currently a lot of academic interest in both the web's content and associated technology because of the web's economic and cultural relevance. Research into web pages itself has been driven by the desire to understand what is available, how it is structured, and how it interacts to other key human activities in order to develop better information retrieval tools, such as search engines. Although advanced functionalities are offered by search engines like AltaVista and InfoSeek, user worries about their dependability have led to the development of a specialized web spider/analyzer to get the raw data by performing direct searches and in-depth analyses of the questioned websites. Information scientists and others who want to will need a web search or web-search-based tool.

Fundamentals of a Web Search

Despite the numerous applications for Web Search, at the core they are all fundamentally the same. Following is the process by which Web Search work:

1. Download the Web page.

2. Parse through the downloaded page and retrieve all the links.

3. For each link retrieved, repeat the process.

Now let's look at each step of the process in more detail.

In the first step, a Web Search takes a URL and downloads the page from the Internet at the given URL. Oftentimes the downloaded page is saved to a file on disk or put in a database. Saving the page allows the Search or other software to go back later and manipulate the page, be it for indexing words (as in the case with a search engine)

or for archiving the page for use by an automated archived.

In the second step, a Web Search parses through the downloaded page and retrieves the links to other pages. Each link in the page is defined with an HTML anchor tag similar to the one shown here:

```
<A
HREF="http://www.host.com/directory/file.html"
>Link</A>
```

After the Search has retrieved the links from the page, each link is added to a list of links to be searched.

The third step of Web Searching repeats the process. All Search work in a recursive or loop fashion, but there are two different ways to handle it. In our project the Links can be Searched using breadth-first manner.

2. Related Work

In this section we will describe the assumptions that are used in the proposed paper.

2.1 Motivation

The economic and cultural importance of the web has guaranteed considerable academic interest in it, not only for affiliated technologies, but also for its content. Research into web pages themselves has been motivated by attempts to provide improved information retrieval tools such as search engines, but also by the desire to know what is available, how it is structured and to determine its relationship with other meaningful human activities. The advanced facilities available in search engines such as AltaVista and Info seek, but their use has raised questions of reliability that have led to the creation of a specialist web spider/analyzer to produce the raw data by direct Searching and analysis of the sites concerned. Information scientists and others wishing to perform data mining on large numbers of web pages will require the services of a web Search or web-Search-based tool, either individually or collaboratively.

The potential power of web mining is illustrated by one study that used a computationally expensive technique in order to extract patterns from the web and was powerful enough to find information in individual web pages that the authors would not have been aware of .A second illustration is given by the search engine Google, which uses mathematical calculations on a huge matrix in order to extract meaning from the link structure of the web. The development of an effective paradigm for a web-mining Search is, therefore, a task of some importance.

A web Search, robot or spider is a program or suite of programs that is capable of iteratively and automatically downloading web pages, extracting URLs from their HTML and fetching them. A web Search, for example, could be fed with the home page of a site and left to download the rest of it.

3. Proposed Algorithm and Methodology

In this paper we are going to implement BFS (Breadth First Search) Algorithm for implementing the search traversal technique and finally the sorted URLS will be placed in the order of BFS Hierarchy where there will be no chance of repetance of visited URL to be replaced once again in between the searched URL's.

3.1 Breadth First Search

Breadth First Search, is an uninformed search method that aims to expand and examine all nodes of a graph systematically in search of a solution. In other words, it exhaustively searches the entire graph without considering the goal until it finds it. It does not use a heuristic.

From the standpoint of the algorithm, all child nodes obtained by expanding a node are added to a FIFO queue. In typical implementations, nodes that have not yet been examined for their neighbors are placed in some container (such as a queue or linked list) called "open" and then once examined are placed in the container "closed".

In order to build a major search engine or a large repository such as the Internet Archive, high-performance Search start out at a small set of pages and then explore other pages by following links in a "breadth first-like" fashion. In reality, the web pages are often not traversed in a strict breadth-first fashion, but using a variety of policies, e.g., for pruning Search's inside a web site, or for Searching more important pages first.

Steps for BFS Algorithm are:

1) Put the starting node (the root node) in the queue.
2) Pull a node from the beginning of the queue and examine it.
    a) If the searched element is found in this node, quit the search and return a result.
    b) Otherwise push all the (so-far-unexamined) successors of this node into the end of the queue, if there are any.
3) If the queue is empty, every node on the graph has been examined -- quit the search and return "not found".
4) Repeat from step 2.

3.2 Pseudo code for BFS Algorithm

The below pseudo code clearly represents the BFS algorithm procedure and its working principle.

Pseudo-Code for BFS Algorithm

```
function breadthFirstSearch (Start, Goal) {
    enqueue(Queue,Start)
    while notEmpty(Queue) {
      Node := dequeue(Queue)
      if Node = Goal {
         return Node  // the code below does not get
executed
      }
      for each Child in Expand(Node) {
        if notVisited(Child) {
          setVisited(Child)
          enqueue(Queue, Child)
        }
      }
    } }
```

4. Implementation Roles

The following are the roles that are performed during the process of crawling the web pages based on individual page traffic**.**

4.1 Google API Role Flow

In this Google API's Role, it shows that operations performed by the Google API after accepting the URL from the user. The default actions are specified by the Ranking Model Web Crawler which is clearly shown in figure 1.

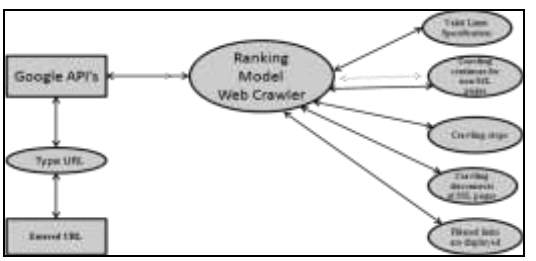


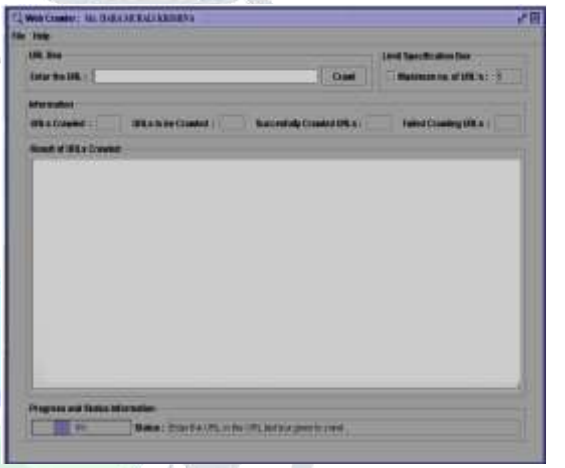
Figure 1. Role of Google API in our application

4.2 Role of User and his Flow of Events

In this User Role, it shows that the operations performed by the user after accepting the appropriate URL details from Google API's**.** The default actions which should be performed by the user is specified by the Ranking Model Web Crawler.

5. Results and Description

For developing this web page extractor tool based on URL traffic, we use java technology with front end User interface designed with java Swings and back end is internet connection for crawling the URL's lively.



6. Conclusion

We have described the architecture and implementation details of our Searching system. Searching forms the backbone of applications that facilitate Web information retrieval. The web Search is designed using Breadth-first Searching, which provides the highest page rankings. Itself capable of searching a large collection of web sites by using idle processing power and disk space. The testing of the system has shown that it cannot operate fully automatically for tasks that involve searching entire web sites without an effective heuristic for identifying duplicate pages.

7. References

1. Herbert Schildt Java Complete Reference. Fifth Edition.

2. Core Java 2: Volume I & II - Fundamentals By Cay S. Horstmann, Gary Cornell

3. Java™ Tutorial, Third Edition: A Short Course on the Basics By Mary Campione, Kathy Walrath, Alison Huml

4. Data Mining Techniques By Arun K Pujari

5. Flippo Menczer, Gutam Pant, Padmini Srinivasan, Searching the Web.

6. Pankaj Jalote, An Integrate Approach to Software Engineering, 3rd Edition.

7. Searching the Web. Gautam Pant, Padmini Srinivasan and Filippo Menczer3

8. S. Chakrabarti. Mining the Web. Morgan Kaufmann

9. F. Menczer and R. K. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the Web. Machine Learning

10. J. Rennie and A. K. McCallum. Using reinforcement learning to spider the Web efficiently.Morgan Kaufmann, San Francisco, CA, 1999.

11. G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

12. "Efficient Searching Through URL Ordering", Junghoo Cho, Hector Garcia-Molina, Lawrence Page. 7th International Web Conference (WWW 98).