Wikipedia Web Crawler

Surbhi Kanthed[1] and Ankit Nimje[2]

I. Problem Statement

Today we are dealing with enormous amount of structured as well as unstructured data. To retrieve information manually become very time consuming and tedious task. In todays era there are lot of competitor and to remain updated about each every competitors would be impossible without some automated search.

II. Objective

We plan to build and develop an application or web application based on web crawlers which will successfully crawl through World Wide Web and create an Index of most relevant terms for records. Best search engine is the one which provides most relevant results with passed query. Our goal is to develop an application which will scan through all recorded indexed terms and find most relevant site which matches our query.

III. Introduction

According to Wikipedia, A Web crawler, sometimes called a spider or spiderbot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web Indexing (web spidering). Web crawlers are basically used in search engines such as Google, Yahoo, etc. and acts as a backend support for such large search engine giants. They are also used in large database systems with very large datasets such as insurance companies, medical history, university systems, etc. A web crawler is a program or application or robot which browses through Internet in progressive or automated manner. It starts with a list of websites to visit (also called as seeds) and identifies all hyperlinks in the current website and adds them to list of URLs (also called as crawl frontier). Thus, gradually progressing to create an index of terms which will be used for future query searches.

Search Engines such as Google and some web sites use Web Crawlers to update their web contents. Web crawlers make indices of visited web pages thus search engines can process more efficiently.

How does Google search works? In the world wide web, there are more than 6 trillion individual web pages. Google keeps track of these webpages by crawling through them and makes index of these web pages. When an user gives a search query, google engine matches it with its indexed terms using generated algorithms while taking care of number of features such as - spelling, search methods, synonyms, autocomplete, Google instant, etc. A crawler starts exploring from a seed page and covers all the hyperlinks within current page. This process is repeated until sufficient number of pages are crawled or some objective is achieved. As we know web is a dynamic entity with content being updated at rapid rates. Were the web being a static collection of pages, we would have fetched all the data and stored in system repository to use it for longer duration.

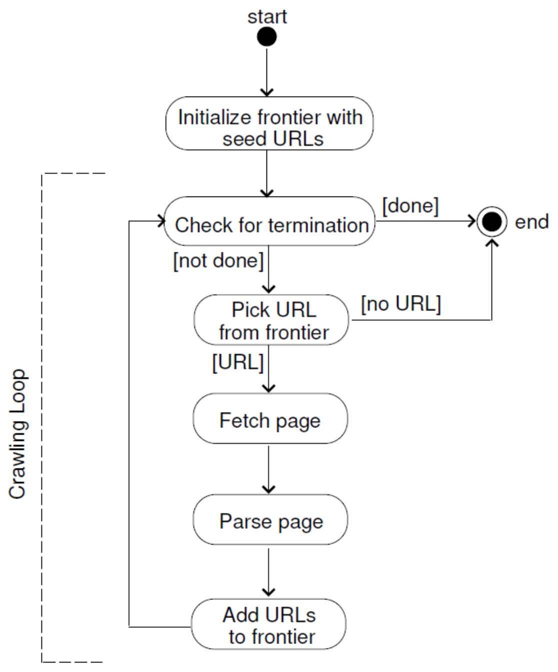

Fig. 1. Flow of sequential Web Crawler

Flow of basic sequential crawler as shown in Figure. As we start, the crawler initializes with list of seed URLs called as frontier. Then starts the crawling loop. It starts with picking the topmost URL from the frontier then fetches it. After parsing it to look for all the hyperlinks in that web page, crawler adds those links into frontier seed list to be fetched. Then a it checks for a termination condition. If the termination condition fulfills it stops or continues crawling cycle. Crawling can be thought as a graph search problem. The Web can be viewed as a large graph with pages as its nodes and hyperlinks a site edges. A crawler starts at a few of the nodes and then follows the edges to reach other nodes.

*This work was not supported by any organization.
*Authors contributed equally.
[1]Masters Student, University of Massachusetts Lowell
[2]Masters Student, University of Massachusetts Lowell

IV. BFS AND DFS WEB CRAWLERS

In DFS, like a traditional DFS technique, starts with first frontier and goes as far as it can go to the depth before backtracking to next same level node. In other words, if we provide a single website as frontier, it will fetch all hyperlinks and crawl very first link found by it. In the same manner our crawler will go to the depth and keep crawling until out of links or if termination condition exists. There is a drawback of traversing using this technique as a crawler can go off topic (out of focus) after series of links and provide unwanted results.

BFS technique as well works like traditional BFS graph algorithm, where it starts with root node (frontier) and traverse through all the neighbor nodes before moving to next level. To explain, for example it starts with one frontier and fetches all links and traverse through all those links and at the same time adding hyperlinks it found in those crawled links to our seed list. Using this technique, there is a small probability of losing focus and going off topic.

V. PAGE RANK ALGORITHM

Page Rank is used by Google Search Engine and also named after Larry Page, one of the founders of Google. It is a way of measuring importance of web pages. According to Google, PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

Page Rank Algorithm is based on following formula:
$PR_{t+1}(P_i) = \sum(PR_t(PR_j/C(P_j))$
Where,
t $\rightarrow$number of iterations
Pj $\rightarrow$Page rank from previous iteration
Pi $\rightarrow$Page rank of current iteration
C $\rightarrow$Number of nodes pointing to current node

Page Rank Algorithm works as a series of iterations with a method defined as pageIterations() with parameters such as:

- G $\rightarrow$a directed graph (all undirected graphs will be converted to directed graphs before passing) Alpha$\rightarrow$damping parameter for page rank
- $MaximumIterations$ $\rightarrow$Number of iterations after which pageIterations() will be terminatednStart$\rightarrow$Starting value of Page Rank for each node.
- $Weight$ $\rightarrow$Weight of edgeConverge$\rightarrow$To check convergence of iterations

Thus, using PR algorithm after all iterations, we get a score for each web page. Higher the score higher the page rank. For instance, to make your webpage display on first few pages of Google Search you should consider provide links to your webpage on many other relevant web pages

VI. CONCLUSIONS

To conclude, we successfully built and developed a web crawler using many features such as Page rank, DFS technique, BFS technique, fetching images, etc. We learnt in detail how a web search engine works. Many challenges were faced while working on this project which is a learning factor.

REFERENCES

[1] S. Chakrabarti, M. Berg, and B. Dom, Focused crawling: a new approach to topic specific Web resource discovery, Computer Networks, vol. 31, pp. 1623-1640, 1999.
[2] W3resource.com Python Flask
[3] Tutorialspoint.com Python Flask
[4] INTRODUCTION OF THE WEB CRAWLING FOR SEARCH ENGINE (ShodhGanga)
[5] Gautam Pant, Padmini Srinivasan, Filippo Menczer - Crawling the Web
[6] Bucky Roberts TheNewBoston
[7] 8] Page Rank Algorithm and Implementation Geeks for Geeks