WORD-RANK BASED WEB CRAWLER

Eyüp Can DÜNDAR
eyupcan@trakya.edu.tr
Computer Engineering Department
Trakya University – Edirne / TURKEY

Aydın CARUS
aydinc@trakya.edu.tr
Computer Engineering Department
Trakya University – Edirne / TURKEY

Altan MESUT
altanmesut@trakya.edu.tr
Computer Engineering Department
Trakya University – Edirne / TURKEY

Abstract

In this work, a word-rank based crawler which makes the link analyze according to the occurrence frequency of the words of a page is developed to show how the different crawling strategies affects the performance. It was seen from the test results that; word-rank based crawler can eliminate the web pages that includes useless data like advertisements. Therefore, users can easily reach the information that they searched without seeing unnecessary links with using a word-rank based crawler.

Keywords: Crawler, Search Engine, Word-rank based crawler.

INTRODUCTION

The World Wide Web has become the biggest digital library. With the introduction of HTML, the web has become the largest accessible store of information. According to Nielsen / NetRatings [1], around the world about over 600 million people have access to the Internet and the average Internet user spends 11 hours and 24 minutes online per month; the average user in United States spends more than twice that amount of time online: 25 hours and 25 minutes at home and 74 hours and 26 minutes at work [2]. The World-Wide Web, having over 11.5 billion pages, continues to grow rapidly at a million pages per day [3]. About 600 GB of text changes every month [4]. Lawrence showed that given the current size of the Web, even large search engines cover only a portion of the publicly available internet; no search engine indexes more than 16% of the Web [5].

Search engines use crawlers to create a copy of the visited pages for later processing that will index the downloaded pages to provide fast searches. A critical problem with search engines is to keep their index up-to-date. While the index continues to grow, more effort is needed to update this index. Since Web documents are dynamic, already stored data becomes useless. Another problem with today's search engines is they are easily manipulated by website owners who could add words repeatedly to their website, even though these words may have had little or nothing to do with the actual subject of their site.

According to The Censorware Project; [6] theoretically, a web crawler starts with a single or a set of default pages and it continues over the hyperlinks until it has downloaded every web page on the World Wide Web. However, it is clear that this theory is not correct in practice. There is not a path from any given page to every other page on the Internet.

Working principle of the crawler is: Crawler starts with a list of pages to visit called the seeds. As the crawler visits these pages it identifies all the hyperlinks in the page and adds them to the list of pages to visit. Crawlers usually perform URL normalization in order to avoid crawling the same resource more than once. The term URL normalization refers to the process of modifying and standardizing a URL in a consistent manner [7].

Crawler must not only have a good crawling strategy but it should also have a highly optimized architecture. While it is clearly easy to build a slow crawler that downloads a few pages

per second for a short period of time, building a high-performance system that can download hundreds of millions of pages over several weeks presents a number of challenges in system design, I/O and network efficiency, and robustness and controllability [8].

Therefore, new studies have made about crawlers like focused crawler that crawls only the one part of the web that focuses on a particular topic [9].

WORD-RANK BASED CRAWLER

Despite rapidly growing World Wide Web, search engines must keep its data up-to-date. Because of that, new studies have made about crawlers.

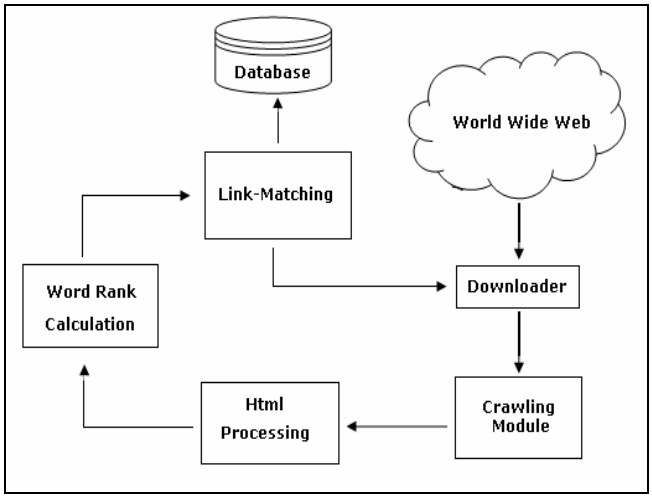

Figure 1. Web Rank Based Crawler Structure

Word-Rank based crawler which is shown in figure 1 has 4 main components.

1. ***Crawling Module:*** This module downloads the web pages which have the value over the word-rank value.
2. ***Html Processing:*** It finds words, links and link texts on the downloaded page.
3. ***Word-Rank Calculation:*** This part finds the word-rank value from the words on the page.
4. ***Link-Matching:*** Links on the downloaded page is matched with the words over the word-rank value and links get a link-value. Links which are over the link-value will be added to link list.

Word-Rank based crawler algorithm is shown in figure 2.

CRAWLING MODULE

In this module, user defines the maximum page count and the first page to crawl. On the program start, first page is appended to link list. Crawler downloads the first page in the link-list and sends the source code of the page to HTML Processing Module. Crawler continues to crawl until it reaches the maximum page count that is defined by user.

HTML-PROCESSING MODULE

This part gets the source code of downloaded page from the crawling module and makes parsing operations and parses the source code to gets the words and links. Stop words that like “are”, “on” in English are not considered by the processing module. Link dictionary for downloaded page is prepared by this module. Link tag has two parts: one of them is the link page address and the other one is the link text or the link image. Html-Processing uses “Regular Expressions”. For “href” tags, regular expression is

<a[^>]*href=(\"|')(.*?)\\1[^>]*>(.*?)

. Program does not only examines href tags, it also examines all possible link tags like frame, iframe, area and basehref tags. Program constitutes link dictionary from the links on the page. Key value in the dictionary is the link text, and the key comment value is the link address.

WORD-RANK CALCULATION

This module gets the word list from html-processing module. Frequently used words get the highest rank on the page. Program sorts the words from frequently used words to rarely used words. Each word gets one point with each of its usage. Word that has the maximum score is on the top of the list. This module sends the word-rank list to Link-Matching module.

LINK-MATCHING

This module gets the link dictionary from html-processing module and ranked word list from the word-rank calculation. The module examines the links, finds the links which has a word that had maximum score in the text of link and continues until all words and all links have been checked.

```
URL  = the page address that has been given by user
Max Page Count = To download maximum page count given by user
Add first URL to Link-List
Do
{
    URL = Link-List First Item
    Delete first item from link-list
    Download the URL
    if (there is at least a link in the page and URL is not processed before)
   {
      Parse the page
      Insert URL to downloaded links table
       Find all possible words on the page
       Make word-list
       Find all links on the page using Regular Expressions
       Make Link-Dictionary
       Give a word-rank value to each word on the word list
       Sort word list maximum rank to minimum rank
       Word-List = Get first 10% items of the word list
       If (count(word-list)>0 and count(link-dictionary)>0)
       {
           For each link in link-dictionary
           {
               For each word in word-list
               {
                    if (link.text includes word)
                    {
                       if (isAbsoluteUrl )
                       {
                           if !(IsHTTPUrl)
                           {
                               Edit link.address
                           }
                          Add link.address to link-list
                       }
                    }
               }
           }
       }
  }
}While (Count(Link-List)>0) and (Max Page Count > Downloaded Page)
```

Figure 2. Web Rank Based Crawler Algorithm

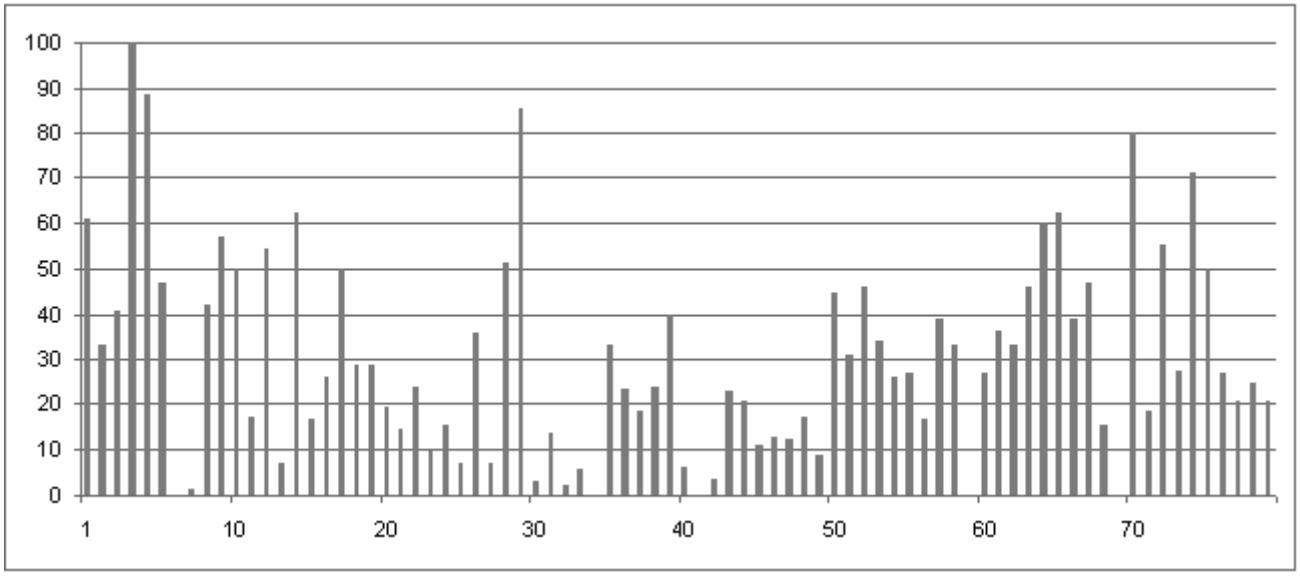

Figure 3. Useful link ratio (%)

EVALUATION

According to the results from the tests on Figure 3, word-rank based crawler continues crawling from the pages that includes links which has information about the page subject.

Word-rank based crawler uses average 30% links on the page. When the page has no links about the page subject, link ratio is zero. Consequently, if page has only advertisements or it is manipulated website by the website owner, word-rank based crawlers do not use the links on that page.

Table 1. *Relevant Pages*

Relevant Pages	Downloaded Pages	Relevance Ratio (%)
99	100	99
100	100	100
99	100	99
79	80	99

Downloaded pages of 99% by the web-rank based crawler have effective information about website subject as seen on Table 1. Thus, search engine will have effective pages on its database and website owners cannot manipulate search engine crawler easily.

Crawlers can retrieve data much quicker and in greater depth than human searchers. Word-rank based crawler uses minimum band-with and minimum forcing on web servers because program downloads pages with in a time period and specific links on the page. Web rank based crawler uses crawler politeness policy [10] that downloads allowed number of pages given by the user.

CONCLUSION

Word-rank based crawlers can be used in efficient search engines, because word-rank crawlers only use specific links about the page subject.

Users can give threshold value for page ratio to indexer that indexes web pages which was downloaded from the crawler to search engines database. If page ratio is over the threshold value, indexer will index the page to database. Thus, search engines only include effective pages that users can easily find information about what they are searched without seeing unnecessary links with using a word-rank based crawler.

REFERENCES

[1] Nielsen//NetRatings, Hitwise, *www.nielsen-netratings.com,* 2003.

[2] http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/execsum.htm.

[3] Gulli, A., Signorini, A., The indexable web is more than 11.5 billion pages, *In Proceedings of The 14th International World Wide Web Conference (WWW)*, 2005.

[4] B. Kahle, Preserving the Internet, *Scientific American*.

[5] Lawrence, S., Giles, C. L., Accessibility of information on the web, 1999.

[6] The Censorware Project www.censorware.org /web_size, 1999.

[7] Lee, S. H., Kim, S. J., Hong, S. H, On URL Normalization*, Proceedings of the International Conference on Computational Science and its Applications*, 2005.

[8] Shkapenyuk, V., Suel, T., Design and Implementation of a High-Performance Distributed Web Crawler*, IEEE International Conference on Data Engineering*, 2002.

[9] Chakrabarti, S., Berg, M., and Dom B., Focused crawling: a new approach to topic-specific web resource discovery*, in Proc. of the 8th International World-Wide Web Conference (WWW8)*, 1999.

[10] Koster, M., Robots in the web: threat or treat?, 1995.