Improved Specific Crawling in Search Engine

K.F. Bharati¹, Prof. P. Premchand², Prof. A Govardhan³

¹Asst.Professor, CSE Dept, JNTUACEA, Anantapur
²Dean, Faculty of Engineering, CSE Dept, UCEOU, Osmania University, Hyderabad
³Director of Evaluation, INTUH, Hyderabad

ABSTRACT

There are many traditional Crawlers, which Crawl the web Pages. But There is a need to find out the specific urls that are related to the specific string in search engine. When a string is entered in specific search engine, It should give the related urls that match the string. Already the research is going in this way to find out the specific urls. The traditional crawler gives only the related urls that match the specific searched string. The proposed work gives related and non-related urls. This improves specific crawler which give more related pages than the earlier crawlers in the search engine.

Keywords: Search Engine, Focused Crawler, Related pages, URL.

1. INTRODUCTION

As the information on the WWW is growing so far, there is a great demand for developing efficient methods to retrieve the information available on WWW. Search engines present information to the user quickly using Web Crawlers. Crawling the Web quickly is an expensive and unrealistic goal as it requires enormous amounts of hardware and network resources. A focused crawler is software that aims at desired topic and visits and gathers only a relevant web page which is based upon some set of topics and does not waste time on irrelevant web pages. Search engines working procedure is as follows. First by sending out a robot to obtain various documents. Indexer then reads these documents and creates an index based on the words, which contained in every document. Proprietary algorithm is used by every search engine. It can create its indexes. Only meaningful results are returned for each query. Search engine is designed and optimized in accordance with domain data. The focused crawler of a search engine aims to selectively search for out pages that are related to a predefined locate of topics, rather than to make use of all regions of the Web. This focused crawling method enables a search engine to operate efficiently within a topically limited document space. The basic procedure of running a focused crawler is as follows. The focused crawler does not collect all web pages, but selects and retrieves only the relevant pages and neglects those that are not concern. But we see, there are multiple URLs and topics on a single web page. So the complexity of web page increases and it negatively affects the performance of focused crawling because the overall relevancy of web page decreases. A highly related area a web page can be hidden. It is because of small on the whole of significance of that page. Apart from main content blocks, the pages have such blocks as navigation panels, copyright and privacy notices, unnecessary images, extraneous links, and advertisements. Segmenting the web pages into small units will improve the performance. A content block is supposed to have a rectangle shape. Page segmentation transforms the multi-topic web page into several single topic context blocks. This method is known as content block partitioning.

The structural design of a general Web search engine is as follows. First it contains a front-end process and a back-end process, as shown in Figure 1. There are several functionalities occurred in every process. The front-end process responsibility is where the user enters the search specific words in the web search engine interface. It is usually a web page and contains a input text box. Then the application parses the search request into a form. That the web search engine understands. Then the it executes the search operation on the index files. Ranking functionality is done in the search engine. After completing the ranking the search engine interface returns the search results to the user. so far front end process functionalities are explained. Next the back-end process functionalities are discussed. A crawler fetches the Web pages from the Internet. And then the indexing subsystem parses the Web pages and stores them into the index files.
A web crawler is an automated script that scans through Internet pages. After it creates an index of the data, there are some uses for the program, maybe the majority accepted being search engines using it to give web surfers with related websites. Crawler searches the web and browses the web. It is for the web Indexing. Crawler is one of the most critical elements in a search engine. It traverses the web by following the hyperlinks and storing downloaded documents in a large database that will later be indexed by search engine for efficient responses to users’ queries. Crawler is one of the most critical elements in a search engine. It traverses the web by following the hyperlinks and storing downloaded documents in a large database that will later be indexed by search engine for efficient responses to users’ queries.

Crawlers are designed for different purposes and can be divided into two major categories. High-performance crawlers form the first category. As the name implies, their goal is to increase the performance of crawling by downloading as many documents as possible in a certain time. They use simplest algorithms such as Breadth First Search (BFS) to reduce running overhead. In contrast, the latter category doesn’t address the issue of performance at all but tries to maximize the benefit obtained per downloaded page. Crawlers in this category are generally known as focused Crawlers. Their goal is to find many pages of interest using the lowest possible bandwidth. They attempt to focus on a certain subject for example pages in a specific topic such as scientific articles, pages in a particular language, mp3 files, images etc.

Focused crawlers look for a subject, usually a set of keywords dictated by search engine, as they traverse web pages. Instead of extracting so many documents from the web without any priority, a focused crawler follows the most appropriate links, leading to retrieval of more relevant pages and greater saves in resources. They usually use a best-first search method called the crawling strategy to determine which hyperlink to follow next. Better crawling strategies result in higher precision of retrieval. Most focused crawlers use the content of traversed pages to determine the next hyperlink to crawl. They use a similarity function to find the most similar page to the initial keywords that is already downloaded and crawl the most similar one in the next step. These similarity functions use information retrieval techniques to assign a weight to each page so that the page with the highest weight is more likely to have the most similar content. There are many examples of focused crawlers in the literature each trying to maximize number of relevant pages.

The focused crawler has three main components:

a) **Classifier**: Classifier, which makes significance judgments on pages, crawled to choose on link extension.

b) **Distiller**: Distiller, which determines a compute of centrality of crawled pages to decide visit priorities.

c) **Crawler**: Crawler, with dynamically reconfigurable main concern controls which is controlled by the classifier and distiller.

### 2. RELATED WORK

Shark-search algorithm [2] was proposed. In this starting URLs, which are relevant to an interested topic to the crawler. Similar to focused crawler a user has to define some starting URLs to the crawler [3]. Hence, the user must have background knowledge about the interested topic to be able to choose proper starting URLs. Crawler does not need any suitable starting URLs, and the crawler can learn its way into the appropriate topic by starting at non-related web pages [6]. InfoSpider, addressed that the user first gives keywords describing the topic of interest to the crawler [4]. Then, the crawler looks for candidate URLs using a search engine, and uses them as the starting point. An advantage of this approach is that the user does not require any background knowledge about the topic but can describe it in terms of keywords in anyway.

Recent works of presented the way to find the starting URLs using a web directory instead of a search engine [5]. Extracting the starting URLs from the web directory will give URLs categorized by a group of specialists. However,
there is a disadvantage when the user’s topic of interest is not in any category of the web directory. In this case, using the search engine seems to be a useful way. There are two basic approaches to specify user interest in topical crawler: taxonomy-based and keyword-based. In taxonomy-based approach, users select their interest from topics of a predefined taxonomy. This approach is simple for users to select their topics. However, it put a great limitation on the set of possible topics. In keyword based approach, interest is specified by keywords that define the targets of the interest. It is more flexible than taxonomy-based approach. However, users might not know how to specify their queries precisely and sometimes even might not be clear about what their true targets are, especially when they are not familiar with the domain of their interest.

A work for designing a search engine which searches specific urls is presented [1]. This approach is to selectively look for the pages to are related to a pre defined group of topics. It does not collect and index web documents. A topic specific crawler analyses its edge limit to discover the links that are the most relevant for the crawl. This leads to major savings in hardware and network resources, and helps remain the crawl more. This work does not give full information about the irrelevant pages. It only specifies the related pages in the all crawled pages. This work specifies graph representation for total relevant and total crawled urls. The proposed approach will give full clarification about the relevant and irrelevant pages.

3. PROPOSED WORK

The proposed Crawler mainly works to develop a system that gives relevant and non relevant pages. This includes time taken to execute the pages also. Here in this system as shown in the figure 2, the relevant and non relevant pages are found. The modules used in the system are described below.

- **User Interface:** User Interface will give the options to begin and stop the crawling process and to enter the URL to crawl and shows the result.
- **HTML Reader:** This module will look the entered URL by the user and reads
- **HTML Parsing:** Html Parsing will recognize markup and separate it from plain text in HTML documents.
- **Check Links:** This module will help to separate good links and bad links.
- **Execution Time:** This Module will give the overall time for execution of good and bad links.
- **Searching:** This module is specific for the searching relevant to the user queries.
The Experiments are conducted on the proposed approach. It gives several results as shown in the graph. Figure 3 shows the graph between total no of crawled urls and related urls found. The results are conducted with the good broadband speed. The approach gives several improvements in the results of finding the related URLs over a specific search.

As the Figure 3 shows graph between the total no of urls crawled and Related URLs found. The above figure 4 represents the results they come in the graph that is between the total no of Crawled URLs and Non-Related URLs.

<table>
<thead>
<tr>
<th>SNO</th>
<th>Total No Of Crawled URLs</th>
<th>Execution Time(Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>62</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>169</td>
</tr>
<tr>
<td>3</td>
<td>250</td>
<td>504</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>771</td>
</tr>
</tbody>
</table>

The Table 1 shows the execution time for crawling the total no of urls.
4. CONCLUSION

Specific Crawlers are the most important trend in these days. Finding the related urls is more difficult in search engine. The traditional specific crawlers give less no of related urls. It consumes more time. The improved specific Crawler gives better results than the earlier specific crawlers. This approach takes the minimum time. Improved Specific Crawler also gives the Non-Related pages also. This work can be extended further in many ways. There is a need to go through this work in the mobile applications. And also it has to be verified in E-commerce applications.

REFERENCES

[1]. B.H.Chandrashekar, Dr.Shobha.G.” Semantic Domain Specific Search Engine” 2010 IEEE

AUTHORS

K.F.Bharati, Asst. Prof., Dept. Of CSE, JNTUACEA, Anantapur, B. Tech From University of Gulbarga M. Tech from Visveswariah Technological University, Belgaum Officer In charge for Central Computer Center, JNTUACEA

P. Premchand Prof, Dean, Faculty of Engineering, CSE Dept. UCEOU, Osmania University, Hyderabad. B. Sc(Electrical Engineering), RIT., Jamshedpur M.E(Computer Science), Andhra University PhD(Computer Science & Systems Engineering), Andhra University, Visakhapatnam

Prof. A. Govardhan, Director of Evaluation, JNTUH, Hyderabad He has done BE in Computer Science and Engineering from Osmania University College of Engineering, Hyderabad in 1992. M.Tech from Jawaharlal Nehru University (JNU), Delhi in 1994 and his PhD from Jawaharlal Nehru Technological University, Hyderabad (JNTUH) in 2003. He was awarded with “NATIONAL INTEGRATION AWARD by HEALTH CARE INTERNATIONAL” and “Dr SARVEPALLY RADHAKRISHNA by A.P STATE CULTURAL AWARENESS SOCIETY” in 2013.In 2012 He was awarded as “The Best Teacher “. He was awarded Telugu Vignana Parithoshikam, by the Government of Andhra Pradesh for B.E(CSE) program. He has been a committee member for various International and National conferences including PAKDD2010, IKE10, ICETCSE-2010 ICACT-2008, and NCAI06.