UDBFC: An Effective Focused Crawling Approach Based On URL Distance Calculation

Debashis Hati
Assistant Professor, School of Computer Engineering
KIIT University Bhubaneswar, India
d_hat@yahoo.com

Amritesh Kumar
Research Associate, School of Computer Engineering
KIIT University Bhubaneswar, India
amritesh.kiit@gmail.com

Abstract—Vertical search engines use focused crawlers as their key component and develops some specific algorithms to select web pages relevant to some pre-defined set of topics. Therefore, to effectively build up a semantic pattern for specific topics is extremely important to such search engines. Crawlers are software which can traverse the internet and retrieve web pages by hyperlinks. Here we propose an UDBFC (URL Distance Based Focused Crawler) algorithm based on a double crawler framework (an experimental crawler and a focused crawler). The main motive of our UDBFC is to measure the relevancy between seed page and child page by vector space model. Seed pages are the common search result generated by three most popular search engine Google, Yahoo and MSN search. Child page links are out links of seed page which are extracted by link extractor tool from seed page. Seed page and child page are fetched by experimental crawler. It calculates the relevancy between seed page and its all child pages. After relevancy calculation it defines groups based on relevancy score. It uses the focused crawler to fetch topic specific pages from internet based on distance score which is calculated between grouped URLs and each URL which is to be fetched.

Keywords—vertical search engine; focused crawler; vector space model; distance calculation.

I. INTRODUCTION

The enormous growth of the World Wide Web (WWW) in recent years has made it important to perform resource discovery efficiently. Consequently, several new ideas have been proposed; among them a key technique is focused crawling which is able to crawl particular topical portions of the WWW quickly without having to explore all web pages. A Web Crawler searches through all the Web Servers to find information about a particular topic. However, searching all the Web Servers and the pages, are not realistic, given the growth of the Web and their refresh rates [5]. Crawling the Web quickly and entirely is an expensive, unrealistic goal because of the required hardware and network resources. Focused Crawling is designed to traverse a subset of the Web to gather documents on a specific topic. It also aims to identify the promising links that lead to target documents, and avoid off-topic searches. Focused Crawlers support decentralizing the crawling process, which is a more scalable approach.

Crawlers are software which can traverse the internet and retrieve web pages by hyperlinks. In the large area of websites, traditional web crawlers cannot function well to solve this problem. The focused crawler of a special-purpose search engine aims to selectively seek out pages that are relevant to a pre-defined set of topics, rather than to exploit all regions of the Web. Focused crawlers aim to search only the subset of the web related to a specific topic, and offer a potential solution to the problem [2, 4]. Focused crawler is developed to collect relevant web pages of interested topics form the Internet. The general-purpose search engines, such as Google (www.google.com), have provided us with a lot of facilities, and become very popular. However, they have some disadvantages because a general-purpose search engine aims to cover the network as enough as possible. So, it usually returns many web pages users are not interested in. Vertical search engines use focused crawlers as their key component and develops some specific algorithms to select web pages relevant to some pre-defined set of topics. Therefore, it is extremely important for a search engine how to effectively build up a semantic pattern for specific topics [1].

The traditional process of focused web crawler is to harvest a collection of web documents that are focused on the topical subspaces. Focused crawlers are the main element in building domain specific search engines. They traverse the web collecting only relevant data to a predefined topic while neglecting on the same time off-topic pages [3]. The crawler is kept focused through a crawling strategy which determines the relevancy degree of the web page to the predefined topic and depending on this degree a decision is made whether to download the web page or not [4].

In this paper we address a webpage choosing strategy for vertical web crawling and propose a two-step choosing strategy. First, an experimental crawler travels sample seed websites and their derived (children or linked) websites in order to collect experimental web pages, from which a set of relevant URL are grouped based on pre-defined topics. The relevancy is calculated between seed page and child page of seed page. We put the URL in relevance group and irrelevant URL are grouped based on pre-defined topics. The main motive of our UDBFC is to measure the relevancy between seed page and child page by vector space model. Seed pages are the common search result generated by three most popular search engine Google, Yahoo and MSN search. Child page links are out links of seed page which are extracted by link extractor tool from seed page. Seed page and child page are fetched by experimental crawler. It calculates the relevancy between seed page and its all child pages. After relevancy calculation it defines groups based on relevancy score. It uses the focused crawler to fetch topic specific pages from internet based on distance score which is calculated between grouped URLs and each URL which is to be fetched.

In section II, we present the key algorithm of URL Distance based focused crawler. In section IV, we present the key algorithm of URL Distance based focused crawler. In section V, the test results of our experiment have been represented and in section VI, we have concluded our research work.
II. RELATED WORK

A focused crawler is a program used for searching information related to some interested topics from the Internet. The main property of focused crawling is that the crawler does not need to collect all web pages, but selects and retrieves relevant pages only. Because the crawler is only a computer program, it cannot determine how relevant a web page is [1].

In order to find pages of a particular type or on a particular topic, focused crawlers aim to identify links that are likely to lead to target documents, and avoid links to off-topic branches. However the concept of prioritizing unvisited URLs on the crawl frontier for specific searching goals is not new, and Fish-Search and Shark-Search were some of the earliest algorithms for crawling for pages with keywords specified in the query [2]. In Fish-Search, the system is query driven. Starting from a set of seed pages, it considers only those pages that have content matching a given query (expressed as a keyword query or a regular expression) and their neighborhoods (pages pointed to by these matched pages). Shark-Search is a modification of Fish-search which differs in two ways: a child inherits a discounted value of the score of its parent, and this score is combined with a value based on the anchor text that occurs around the link in the Web page.

There are many approaches to select the strategy for focused crawler. Many researchers have written their approaches based on link analysis. For example, Effective Focused Crawling based on content and link structure analysis has been proposed for link analysis based on URL score, anchor score and relevance score and HAWK: A Focused Crawler with Content and Link Analysis. Some have written their approaches based on page rank value. For example, An Application of Improved Page Rank in Focused Crawler based on To-page rank value and an Improvement of Page Rank for Focused Crawler based on T page rank. Some have written based on ontology. For example, A Survey in Semantic Web Technologies-Inspired Focused Crawlers and A Transport Service Ontology-based Focused Crawler based on ontology. Some have developed based on meta search and content block partition “A Framework of a Hybrid Focused Web Crawler.” Some have developed rule based focused crawler. For example, Design of an Enhanced Rule based Focused Crawler and URL rule based focused crawler.

A working process of a focused crawler is composed of two main steps. The first step is to determine the starting URLs and specify user interest. The crawler is unable to traverse the Internet without starting URLs. The second step in a focused crawling process is the crawling method. In theoretical point of view, a focused crawler smartly selects a direction to traverse the Internet. A clever route selection method of the crawler is to arrange URLs so that the most relevant ones can be located in the first part of the queue. The queue will then be sorted by relevancy in descending order [7]. The performance and efficiency of a focused crawler is mainly determined by the ordering strategy that determines the order of page retrieval.

III. PROPOSED ARCHITECTURE

Seed URLs are extracted by threeSearches.com. Now by using link extractor tool, we find out the all forward links of seed link because the seed page link is most topic relevant link for query. Experimental crawler fetches the web page of seed URLs and all forward links of seed URLs. After fetching all forward link web pages of particular seed URL, we calculate the relevancy score based on vector space model between seed page and all child pages of this seed page based on keyword analysis. Based on relevancy score, we put the child URLs into relevant group or irrelevant group. Seed URL is always put into relevant group. In URL repository only unvisited URLs are stored. Seed URL and its child URLs not fetched by focused crawler because it is already fetched by experimental crawler. To identify which URL is topic relevant and which URL is not, focused crawler takes a URL from URL repository and calculate the distance of this URL to all URLs which has been grouped in relevant group. If distance score of unvisited URL with any relevant grouped URL is less than or equal to 1, then this unvisited URL is topic relevant. Otherwise, it is irrelevant URL. Focused crawler takes this unvisited topic relevant URL and fetches the corresponding web pages from internet. The proposed architecture is shown below in Fig. 1.

A. Seed URLs Extractor

In our proposed approach, seed URLs are extracted by one search engine known as threeSearches.com. We put a
query in this search engine and it shows the result of three most popular search engines like Google, Yahoo, and MSN search. We take resulting URLs which are common in all the three search engines. We assume that this common search result URLs are most relevant for this query and thus these URLs are the seed URLs.

B. Experimental Crawler

In our approach, we use two crawlers: an experimental crawler and a focused crawler. Experimental crawler is used for fetching the seed page and its child pages. All these pages are experimental pages and we calculate the relevancy between the seed page and experimental pages. Based on relevancy score we identify the group.

C. Relevancy Calculation

Relevancy is calculated by vector space model. We remove the stop words and we extract the top weighted 10 terms with its weight from seed page and child page. Based on relevancy score we identify a group. If child page has a relevancy score with seed page more than 0.9, then we put that child page URL in relevant group which are shown in TABLE I and TABLE II (for one seed page and corresponding child page.) The formula for calculating relevancy score is as follows:

\[
R(s, c) = \frac{\sum wks \times wkc}{\sqrt{(\sum wks)^2 \times (\sum wkc)^2}}
\]

where ‘s’ denotes the seed page, ‘c’ denotes the child page of seed page, wks and wkc are the weights of keyword k in the weight table of seed page and in the weight table of child page respectively.

<table>
<thead>
<tr>
<th>Word</th>
<th>Frequency</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>80</td>
<td>0.0307</td>
</tr>
<tr>
<td>Cricket</td>
<td>76</td>
<td>0.0295</td>
</tr>
<tr>
<td>Team</td>
<td>45</td>
<td>0.0197</td>
</tr>
<tr>
<td>Match</td>
<td>33</td>
<td>0.0159</td>
</tr>
<tr>
<td>Inning</td>
<td>29</td>
<td>0.0146</td>
</tr>
<tr>
<td>England</td>
<td>20</td>
<td>0.0118</td>
</tr>
<tr>
<td>Playing</td>
<td>16</td>
<td>0.0105</td>
</tr>
<tr>
<td>May</td>
<td>16</td>
<td>0.0099</td>
</tr>
<tr>
<td>Australia</td>
<td>14</td>
<td>0.0099</td>
</tr>
<tr>
<td>New</td>
<td>14</td>
<td>0.0099</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table II. Child Page Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Test</td>
</tr>
<tr>
<td>Cricket</td>
</tr>
<tr>
<td>Team</td>
</tr>
<tr>
<td>Match</td>
</tr>
<tr>
<td>Inning</td>
</tr>
</tbody>
</table>

D. URL Repository

In URL repository, we put the forward link URLs of relevant page with the help of link extractor tool. To identify relevant URLs, focused crawler take URLs from URL repository and calculate the distance of this URL from each relevant grouped URLs.

E. Distance Calculation

The distance of input URL is calculated with relevant grouped URLs. If the distance score is less than or equal to 1, then input URL belongs to topical relevant URL and focused crawler take this input URL and fetch the page of this URL from internet.

a) Data Structure for URLs. We break an URL into three segments: host, path and query. The host segment is its domain name. The path segment contains a series of directories. The query segment contains a series of key value pairs. Taking the URL “http://www.chinapub.com/member/buybook/view.asp?add=1&tid=203839”, for example, its host is ‘www.chinapub.com’, its path is ‘/member/buybook/view.asp’, which is made of directories ‘member’, ‘buybook’, and ‘view.asp’, and its query is ‘add=1&tid=203839’, whose key-value pairs are (add,1) and (tid,203839).

b) The Distance Metric between URLs. Now let us discuss the distance metrics. Let i, j be two URLs. The distance metrics between URLs i, j can be calculated based on the distance metrics of their corresponding segments.

\[
d_{URL}(i, j) = (d_{HOST}(i, j) + 1) \times (d_{PATH}(i, j) + 1) \times (d_{QUERY}(i, j) + 1)
\]

where \( d_{HOST}(i, j) \) denotes the distance between the host segments of i and j, \( d_{PATH}(i, j) \) is the distance between the path segments, and \( d_{QUERY}(i, j) \) is the distance between the query segments. The three distances are calculated as follows:

1. \( d_{HOST}(i, j) = 32 \), if \( Host(i) \neq Host(j) \), otherwise \( d_{HOST}(i, j) = 0 \). The value \( d_{HOST}(i, j) \) is set to be a big value. The reason we set \( d_{HOST}(i, j) = 32 \) is that according to the result of experiments it is a value which can help our crawler lead to a better situation.

2. Assuming that the paths of i and j have m directories and n directories respectively, and \( m \leq n \). Given that the number of the different directories in the first m places is k, \( d_{PATH}(i, j) = k \times 2 + (n-m) \times 4 \).

3. \( d_{QUERY}(i, j) = 0 \) if Query(i) = Query(j) ; otherwise \( d_{QUERY}(i, j) = 1 \). This is based on the assumption that if two URLs are equal to each other except for the query segments for most websites, their web pages are usually generated by the same template and belong to the same topic.
IV. PROPOSED ALGORITHMS

A. Algorithm 1

- Step 1: Extract page from seed URLs
- Step 2: All links are extracted from seed page by link extractor tool
- Step 3: for seed_page = 1 to total_seed_page
- Step 4: Count total number of child links of each seed page and store this value in total_child_page
- Step 5: for child_page = 1 to total_child_page
- Step 6: if (sim(child_page, seed_page) >= 0.9), put child_page into relevant group
- Step 7: else, put child_page into irrelevant category

In step 1, we extract the page of seed URLs. In step 2, we know that the seed page is most relevant page for query. So, we extract all forward links of seed page and assume that those are also topic relevant. In step 3, we count the total number of seed pages and put into the variable total seed page and we start a loop from 1 to total seed pages. In step 4, we extract all forward links of each seed page and put that value in one variable named as total_child_page. In step 5, we start a loop from 1 to total number of child pages in each seed page. In step 6, we calculate the relevancy score from seed page to each and every child page of particular seed page and we identify a group based on relevancy score. If relevancy score is greater than 0.9, we put this child page URL into relevant group. In step 7, if relevancy score is not equal to or greater than 0.9, we put this child page URL into irrelevant group.

B. Algorithm 2

/*This algorithm is for finding out duplicate URLs in relevant group. We identify this based on distance formula.*/
- Step 1: for i = 1 to total child page
- Step 2: for j = 1 to total child page
- Step 3: if (i <= j)
- Step 4: calculate distance from link i to link j
- Step 5: if (dis(i, j) == 0) break
- Step 6: remove link i from relevant group

C. Algorithm 3

/*This algorithm is used for identifying the relevant link which is to be fetched. To identify the relevant link we calculate the distance score between URL which is to be fetched and relevant grouped URLs.*/
- Step 1: focused crawler take input link URL from URL repository
- Step 2: for fetched_link = 1 to total_relevant_link
- Step 3: if (dis(fetched_link, link) <= 1), break
- Step 4: if (fetched_link == total_relevant_link)
- Step 5: link is irrelevant link

In step 1, focused crawler takes a URL from URL repository. In step 2, to identify that the particular URL is topic related or not, we start a loop from 1 to total number of relevant grouped URLs. In step 3, if distance of link and fetched_link is less than or equal to 1, then the link is topic related link and loop is broken. In step 4, if fetched link value is equal to total_relevant_link, then it means that the link does not have distance value less than 1. It means that it is not topic related.

V. V. TEST RESULTS

For example, we have taken query “test cricket.” We put this query into www.threesearches.com. It shows the result of three most popular search engines Google, Yahoo, and MSN search in one web page. We select the common URLs in all the results and take it as a seed URL because we think that the most relevant URLs are common in all three search engines. Here, www.threesearches.com shows the result for query “test cricket” in all three search engines. One common URL in top 10 URLs is http://en.wikipedia.org/wiki/Test_Cricket. Here we have taken 10 because we think that the most relevant result for query is top 10 results. We take this URL as seed URL. We put this URL as an input in link extractor tool. The output of this tool is all forward links of seed URL. After seed page as well as all child pages are fetched by experimental crawler, we find out top ten weighted terms of seed page and child page.

In TABLE I, seed page table’s URL is http://en.wikipedia.org/wiki/Test_Cricket and in TABLE II, child page table’s URL is http://en.wikipedia.org/w/index.php?title=Test_Cricket&action=edit&section=15. In our tables, we have mentioned the top 10 weighted terms of seed page and the weight of these terms in child page except the “stop” word. Now based on these table values, we calculate the relevancy score of child page and find out that the relevancy score is 0.92418234. Based on this relevancy score, we put this child URL into relevant group because its score is greater than 0.9.

Now, we assume that the focused crawler takes one URL from URL Repository and check it is relevant to topic or not based on distance calculation with relevant grouped URL. For example, focused crawler takes one URL from URL repository which is http://en.wikipedia.org/w/index.php?title=Test_cricket&action=history. Now, calculation of the distance of this input URL is done by focused crawler with the relevant grouped URLs. In our example, one relevant grouped URL is http://en.wikipedia.org/w/index.php?title=Test_Cricket&action=edit&section=15.
Here, the host name of input URL is same with the host name of relevant grouped URL. So, \( d_{\text{HOST}}(i, j) = 0 \). Its path name is also same. So, \( d_{\text{PATH}}(i, j) = 0 \) but query name is different. So, \( d_{\text{QUERY}}(i, j) = 1 \).

\[
d_{\text{URL}}(i, j) = (d_{\text{HOST}}(i, j) + 1) \ast (d_{\text{PATH}}(i, j) + 1) \ast (d_{\text{QUERY}}(i, j) + 1) - 1
\]

Result of distance = 1. So, this unvisited URL is relevant and focused crawler fetches web page of this URL.

**VI. CONCLUSION AND FUTURE WORK**

Focused crawling methods are the important members of the search engine family. But one of the key problems of vertical search engines is to develop an effective algorithm for the topic-specific search and the similarity measurement among topics. One approach for solving this problem is to analyze the retrieved URLs and the URL patterns and their formal representation are discovered. In this paper we propose an UDBFC algorithm to meet this requirement for focused crawling. Distance calculation is used for calculating the distance between URLs based on its host name, path name and query similarity. UDBFC is based on distance calculation between visited URLs and the URLs which are to be fetched.

Thus our future work will focus on improving the URL regular expression learning algorithm and making the crawler automatically finds topic specific websites in entire web to obtain an URL from the whole internet using distance based focused crawler. We expect to do more extensive test later using Java API.

**REFERENCES**


