# Implementation of Hybrid Approach for Page Relevance Computation

Kompal Aggarwal[1] ,Rajender Nath[2],Giridhar Gopal Bansal[3]
1 Department of Computer Science & Applications
Kurukshetra University, Kurukshetra
2 Department of Computer Science & Applications
Kurukshetra University, Kurukshetra.
3 Department of Computer Science & Applications
S.D College,Ambala Cantt

**Abstract**
The Focused crawlers aim is to find out only the relevant pages which are related to some specific topic. The Topic specific weight table defines the crawling target. The weight of the page corresponding to each keyword in the topic specific weight table is calculated. The topic specific weight table can be created in many ways. The Crawler makes the decision regarding downloading of page or not from the Page weight. This paper proposes HAPRC model for page relevance calculation. The experimental result demonstrates that the proposed HAPRC model achieved a significant improvement over related state-of-the-arts approaches.

**Keywords**: Relevance, Focused, Crawler, Hybrid

## 1. Introduction

World Wide Web (WWW) contains a huge amount of information and with every second of clock latest information is updated such that the web size is of order tens of billions of pages. Web crawler is one of the key component of search engine. It continuously downloads webpages and these pages stored in database after indexing. So, it becomes not so easy for a crawler to crawl the complete web and keeps updated index. Because of inadequacy of various computing resources and time constraints, focused crawlers came into picture.

A focused crawler ideally downloads only those web pages that are relevant to a predefined topic and avoid downloading all irrelevant pages.[1] Therefore a focused crawler detects that a link to a certain webpage is relevant before truly downloading the page. The performance of a focused crawler mainly depends on the quality of links[2] in the particular topic being searched and focused crawling generally relies on a general search engine for giving starting points[3][4]. As the content of webpages is changing very fast it has become a requirement to develop a system which can download relevant Web pages in minimum browsing time. During downloading a Webpage, the sequence of accepting the URL, fetching the page, parsing the page, extracting all the hyperlinks is performed [5]. Hence, while downloading the page the bandwidth is used up anyways. It will be even more beneficial if we utilize the used bandwidth. The downloaded pages bandwidth can be used to get the title, body and the number of outgoing links on that particular page.

## 2. Problem Formulation

[6] proposed a method for focused crawling that permits the crawler to go through several irrelevant pages to search for the next relevant one when the current page is irrelevant .The author uses topic specific weight table in which Keywords weights are computed once statically based on term and document frequency. [7] proposed a method for retrieving relevant pages in which topic specific weight table is constructed in dynamic way i.e. based on frequency of the term in the queries which are stored in the query log.
The main problems of the above mentioned approaches are that topic specific weight table is either static based on occurrence of terms in the documents or dynamic based on occurrence of terms in the input queries. In static construction, the weight table does not change due to change in input queries and in dynamic construction the weight table does not change due to change in the documents in which term appears.

## 3. Proposed HAPRC Model

To address the problems stated above, the hybrid approach for page relevance computation (HAPRC) model for construction of topic specific weight table based on term frequency, document frequency and query frequency is proposed. After construction of topic specific weight table,the relevance of webpage is computed using page weight which is sum of weight of the URL, title and body. The proposed model intelligently decide about page relevance using various parameters and skip non relevant pages.

The proposed model starts with a seed URL and not downloads the page, instead it parse the page for extracting the URLs and significant words into that page.
To determine the significance of the page being parsed, the topic specific weight table is constructed in a hybrid way. Topic is defined as the set of

keywords along with their weights associated with them.
Suppose T is considered as topic and the topic vector can be written as:

$T = [(k_1,w_1), (k_2,w_2), (k_3,w_3),\ldots. (k_j,w_j)]$

where $k_j$ represent $j^{th}$ keyword or phrase of topic T.
$w_j$ is the weight of the $j^{th}$ keyword and represent importance of the $j^{th}$ keyword in the topic T.
To construct topic Specific Weight Table (TSWT),the term weight is computed as

$W_i^{new} = (1 - \alpha)$ qf + (1- α) (tf*df) + $W_i^{old}$ … …… (3.1)

α is taken as a constant and has value stands between 0 and 0.5
The term frequency(tf) is appearance of term t in top n relevant documents and is calculated as follows

$tf=d_1+td_2+$----------$+td_n$ .....(3.2)

$td_1,td_2\ldots.td_n$ are frequency of occurrence of term t in documents $d_1,d_2\ldots\ldots\ldots\ldots d_n$.
The document frequency(df) is frequency of appearance of term t in number of documents(N ) i.e

df = N.....(3.3)

The query frequency(qf) is frequency of appearance of term t in queries existing in the query log and is calculated as follows.

$qf=nq_{1+}nq_2+$----------$+nq_n$ .....(3.4)

$nq_1,nq_2,$.....,$nq_n$ are frequency of occurrence of term t in queries $q_1,q_2,\ldots..q_n$ existing in query log.

The terms having weight greater than or equal to certain defined threshold value can be taken as keyword to be added in TSWT and used for knowing the Page relevancy.
After construction of TSWT, the page relevancy is calculated on the basis of weight given to the page. The Page weight is calculated on the basis of appearance of every keyword in TSWT in different parts of a page by using following equation.

kwp=kwurl+kwt+kwb.....(3.5)

kwp is weight of keyword k in page p
kwurl is keyword k weight based on page URL
kwt is keyword k weight based on title of page
kwb is keyword k weight based on body of the page

As appearance of same words at different locations of a page has different importance and representing various kind of information. So, page relevance which is being parsed can be decided by considering each of the component. For example, the title text is more important for expressing the topic covered in a page as compared to the common text. If value of page weight crosses predefined threshold value, only then the page will be downloaded, otherwise the page will be discarded. In this way we save a lot of bandwidth after discarding an irrelevant page and network load is reduced.

**Pseudo-Code of HAPRC Model:**

Begin

(I). Input user query.
(II). Add seed URL to queue.
(III). While (queue is not empty)

(i) Pick URL from queue
(ii) Fetch the page p and parse it.
(iii) Split the query into terms.
(iv) Retrieve top n relevant documents using terms retrieved in step (iii).
(v) For each term, found in step (iii) calculate tf, df, qf

tf= td1+td2+----------+tdn
//$td_1,td_2\ldots..td_n$ are frequency of occurrence of term t in documents d1,d2……dn.
df = N
//document frequency(df) is frequency of occurrence of term t in number of documents(N).
$qf = nq_{1+} nq_2+$----------$+nq_n$
// $nq_1,nq_2,$.....,$nq_n$ are frequency of occurrence of term t in queries $q_1,q_2,\ldots..q_n$ existing in query log.
$W_i^{new} = (1 - \alpha)$ qf + (1- α) (tf * df) + α $W_i^{old}$
//$W_i^{new}$ is the new term weight and α is taken as a constant whose value stands between 0 and 0.5. $W_i^{old}$ is the old term weight if that term appear in previous weight table constructed if not occurred before then it taken as 0 and $W_i^{old}$ is the term current weight.
If $W_i^{new}$ > threshold value then
add ith term to the topic specific weight table and goto step III(v)
endif
endfor

(vi) For each keyword in topic specific weight table

If(keyword present in URL) then
wkurl =furl*wurl
endif

If (keyword present in title of page p) then
wkt = ft*wt
endif
If (keyword present in the body of page p) then
wkb =fb* wb
endif
wkp =wkurl+wkt+wkb
If (wkp > threshold_value)
Download Page p.
endif
endfor

endwhile

End

## 4. Experimental Setup & Results

For checking effectiveness efficiency, the proposed HAPRC model is programmed in Python 3.0. In this experiment, a corpus of 3204 documents from CACM is taken. The corpus.txt has document id begins with # followed by document contents that are already stemmed .

portabl oper system
portabl oper parallel algorithm
appli stochast parallel algorithm
portabl oper appli stochast
portabl oper parallel stochast
parallel algorithm oper
equat numer techniquear present for comput the root
numer represent of entir word or common phrase
recommend graduat school of comput scienc
convers between float point represent
numer solut of the polynomi equat algorithm
solut of polynomi equat by bairstow
euler summat algorithm
mullers method for find root of an arbitrari function algorithm
numer control machin tool propos american standard
a model for autom file and program design in busi applic system
some thought on reconcil variou charact set
portabl oper appli
parallel algorithm oper equat numer
recommend graduat school euler summat algorithm
euler summat numer control machin
polynomi equat american standard
polynomi equat for autom file
portabl oper polynomi equat
numer represent equat algorithm
autom file variou charact set
hitchcock agrau commun
appli stochast oper portabl
symbol manipul by thread list

The outcome of the execution of the model for all input queries is delineated in table 4.1 given below. Further, in the results "+++++" indicates that the page has been downloaded while "-------" indicates that the page has not been downloaded.

**Table 4.1 : Outputs for different input queries of HAPRC Model**

| Input Query | Output |
|---|---|
| portabl parallel algorithm | +++++Download the document no. , '1930'<br><br>+++++Download the document no. , '3127'<br><br>+++++Download the document no , '2246'<br>-------- Don't Download the document no, '3196'<br>+++++ Download the document no. , '2714'<br>+++++Download the document no. , '2973'<br>+++++ Download the document no. , '2433' |
| parallel oper appli stochast | +++++ Download the document no. , '1696'<br><br>-------- Don't Download the document no. , '1194'<br><br>+++++ Download the document no. , '1811'<br><br>+++++ Download the document no. , '3059'<br><br>+++++ Download the document no. , '1262'<br><br>--------- Don't Download the document no. , '1008<br><br>++++++ Download the document no. , '2342' |
| float point represent | ++++++ Download the document no. '2525'<br><br>++++++ Download the document no. '1843'<br><br>++++++ Download the document no. '1634'<br><br>-------- Don't Download the document no. '183'<br><br>+++++ Download the document no. '1705'<br><br>-------- Don't Download the document no. '628'<br><br>+++++ Download the document no. '3131' |

| | |
|---|---|
| solut of polynomi equat | +++++Download the document no. , '1387'<br><br>+++++Download the document no. , '111'<br><br>--------Don't Download the document no. , '342'<br><br>--------Don't Download the document no. , '647'<br><br>--------Don't Download the document no. , '325'<br><br>-------- Don't Download the document no., '112'<br><br>+++++Don't Download the document no. , '1599' |
| mobile crawler network load | ++++++ Download the document no. , '2951'<br><br>++++++ Download the document no. , '2712'<br><br>++++++ Download the document no. , '2892'<br><br>++++++ Download the document no. , '2849'<br><br>'++++++ Download the document no.', '1685'<br><br>++++++ Download the document no. , '2776'<br><br>++++++ Download the document no. , '2860' |
| focused crawler relevance | ----------' Don't Download any document' |



**References**

[1] Dvijesh Bhatt, Daiwat Amit Vyas and Sharnil Pandya, "Focused Web Crawler", Advances in Computer Science and Information Technology (ACSIT), Vol. 2(11), pp. 1-6, Apr-Jun 2015.

[2] Satwinder Kaur1 & Alisha Gupta "A survey on web focused Information extraction Algorithms", international journal of research in computer applications and robotics, Vol.3 Issue.4, Pg.: 19-23 ,April 2015

[3] Anish Gupta and Priya Anand, "Focused web crawlers and its approaches", International Conference in Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Noida, IEEE, pp. 619-622, 25-27 Feb 2015

[4] S. Mali and B.B. Meshram, "Focused web crawler with revisit policy", In Proceedings of the International Conference & Workshop on Emerging Trends in Technology, New York, ACM, pp. 474-479, 25-26 Feb 2011

[5] Dvijesh Bhatt, Daiwat Amit Vyas and Sharnil Pandya, "Focused Web Crawler", Advances in Computer Science and Information Technology (ACSIT), Vol. 2(11),

pp. 1-6, Apr-Jun 2015

[6] Anshika Pal, Deepak Tomar, S. Shrivastava(2009), “Effective Focused Crawling based on Content & Link Structure Analysis” Vol. 2, No. 1, June 2009

[7] Meenu, rakesh batra, "A review of focused crawler approaches", ijarcsse volume 4, issue 7, july 2014