A Filtered-Page Ranking
An Approach for Previously Filtered HTML Documents Ranking

Jose Henrique Calenzo Costa and Carina F. Dorneles

Federal University of Santa Catarina
Technological Center
Informatics and Statistics Department
Florianpolis - SC, Brazil
Email: henriquecalenzo@gmail.com, dorneles@inf.ufsc.br

***Abstract*—This paper describes a ranking approach applied over previously filtered documents, which relies on a segmentation process. The ranking method, called Filtered-Page Ranking, has two main steps: (i) page segmentation and irrelevant blocks removal; and (ii) document ranking. The focus of the first step is to eliminate irrelevant content from the document, which has no relevance to user query, by means of the Query-Based Blocks Mining algorithm, creating a filtered document that is evaluated in the ranking process. During the ranking step, the focus is to calculate the relevance of each filtered document for a given query, using criterias that prioritizes specific parts of the document and to the highlighted features of some HTML elements. As shown in our experiments, our approach outperforms the base line Lucene implementation of vector space model. In addition, the results demonstrate that our irrelevant content removal algorithm improves the results and our relevance criterias make difference to the process.**

Keywords–Page segmentation; HTML Ranking; Web content automatic extraction; Irrelevant content removal.

I. INTRODUCTION

The process of `ranking` documents is part of many applications, such as search engines [1][2][3], recommendation systems [4][5][6][7], document classification [8][9], among others [10][11][12]. The focus of approaches varies and usually defines different relevant parameters for the ranking. In general, the ranking process of documents has been treated traditionally as a matching problem between a query and a set of documents. In this context, a common challenge is to find a way to select representative documents to a specific query and to explore new ranking models that produce accurate results.

HTML documents ranking algorithms can be built by taking into account several aspects. Selvan et.al [13] propose three categories of ranking algorithms: (i) based on links analysis, which focus on links analysis of a document set to define the ranking; (ii) based on custom search, which considers the users' query or the feedback aspects provided by them; and (iii) based on page segmentation, which consists of algorithms that divide the page into blocks. We propose an approach that uses features from the three categories since considers the users' query on a fragmented document analysing the links on it. Beside that, the ranking function uses some parameters that consider most relevant those documents that have the query terms in key blocks such as main title, first sentence of paragraphs, highlighted sentences, etc. In the literature ranking algorithms, the existing approaches use the whole document in the process [1][2][3]. The problem is that, usually, we are interested only in the content regions that contain the query.

This paper describes an approach to rank previously filtered HTML documents, which is user query-based, called Filtered-Page Ranking (FPR). The ranking process has two main steps: (i) irrelevant content removal using page fragmentation; and (ii) documents raking using the filtered (fragmented) page. The intuition behind the process is to rank an HTML page using just its relevant and useful content. For "relevant and useful content" we mean content that is related to the user's query terms. The purpose of the first step is to generate a filtered document containing only user query content, which is evaluated in the ranking stage, through an algorithm called Query-Based Blocks Mining (QBM), which generates a filtered document that is evaluated in the ranking stage. The segmentation is performed based on the terms of user query, on important criteria that consider different documents components, and on some highlighted HTML elements. In order to do that, the documents are segmented into relevant, highlighted and disposal blocks, excluding those one considered irrelevant. During the ranking step, relevance criteria are used to indicate how close the content of a page is to the query terms. The ranking focuses on defining the relevance of filtered HTML pages for a given query. This paper presents the following contributions:

- an algorithm to remove irrelevant content: a user-query based method, that eliminates from the document those blocks that are considered irrelevant since they are not related to the user's query;
- an algorithm to rank segmented and filtered pages: a method that evaluates specific aspects of a document, with different weights, for ranking calculation, such as terms in bold, term occurrences in the title, highlighted terms (section III-A) and so on.

To evaluate our proposal, experiments have been performed on a document repository and the results are compared with the following existing proposals: the vector model, as the ranking algorithm [14], through the implementation of Lucene [15], and the irrelevant content removal algorithm called Content Extraction via Tag Ratios (CETR) [16]. The experiments show that our irrelevant content removal algorithm improves the results, and that the criteria used to calculate the relevance of HTML pages are meaningful in the ranking process.

 ISBN: 978-1-61208-474-9

This paper is organized as follows. In Section II, we present some works related to our proposal. The proposed ranking method is described in detail in Section III, where we show the irrelevant content removal phase and the ranking process itself. The experiments are presented in Section IV, showing the methodology we used and the results achieved. In Section V, the conclusions and the future work are described.

II. RELATED WORK

In this section, we present some related work of ranking methods for HTML pages classified into three different categories [13], which can be built by taking into account several aspects.

Proposals that are based on the links analysis focus on the links analysis of a document set to define the ranking. The classic PageRanking algorithm is an example that uses a ranking technique based on the relationship between several web pages [17] and Hypertext Induced Topic Search (HITS) [18], which was developed to quantify the authority and the hub values of a page. A page has a high authority value when it is pointed by many other pages (hubs) and a high hub value when it points to several other pages (authorities). In this group, the algorithms are often fully automated and very useful for setting the initial ranking of a large set of web pages without a user interaction.

The second category, based on custom search, considers the users' query or the feedback aspects provided by them. In this category, Duhan et al. [19] uses the term Web Usage Mining (WUM) to identify these studies. In this technique, with the user being recognized by the system through information gathering (researches done, pages accessed), pages that may be more important for a particular search than others are found. The proposal of Joachims [20] is to use clickthrough data that specifically uses the information of links accessed (clicked) by the user to make these visited pages the priority. The method called Page Content Ranking (PCR) [21] evaluates the proximity of the web page with the query terms made. It is based on characteristics such as the frequency of terms, the number of pages containing the term and the occurrence of synonyms, comparing PCR with PageRank. The PCR applies a neural network to detect the importance of a page for a particular search, which requires network training and consequently user interaction. Another example of this category is the ranking algorithm of Lucene [22], which uses the Vector Space Model (VSM) or the Boolean model to determine the relevance of a given document in relation to a specific query from a user.

Finally, proposals that are based on page segmentation consist of algorithms that divide the page into blocks. Some works, such as FixedPS [23], Block-Based Web Search [24] and Computing Block Importance for Searching on Web Sites [25], use this approach. The main idea is to divide the document into homogeneous zones, where each one has the same type of content. Considering each block individually can be useful to separate the different kind of content, meanly to increase the ranking process performance.

The method proposed in this paper performs a segmentation process and at the same time considers the users query to improve the ranking and uses link analysis to calculate the page relevance, having similar aspects from all categories. Table I contains some features we use to compare the proposals, considering HTML/WEB specific aspects, personalized ranking and the use of user query for ranking document, user's navigation and the use of artificial intelligence.

font=footnotesize,sc,justification=centering,labelsep=period

TABLE I. WEB RANKING ALGORITHMS.

Algorithms	Particular HTML aspects (links, tags, styles...)	User's query based	User's Navigation	A.I
Page Ranking	Yes	No	No	No
PCR	No	Yes	No	Yes
VSM	No	Yes	No	No
FixedPS	No	Yes	No	No
ClickThrough Data	Yes	No	Yes	Yes
Block-Based Web Search	Yes	Yes	No	No
Block Importance on WebSites	No	Yes	No	No
FPR (our proposal)	Yes	Yes	No	No

Regarding the Block-Based Web Search method, PCR, VSM and Block Importance for Searching on WebSites, these take into consideration general aspects of ranking documents as frequency of terms and reverse frequency of terms, not taking into account the use of html tags for use criteria as highlighted terms, the tag <title> or <meta> (despite the Block-based Web Search perform the ranking of the content contained in the <title> tag only). Block Importance for Searching on Website also does not consider aspects like highlighted terms, if terms appear on the tags <title> and <meta> and this requires many pages using a similar template. The Page Rank does not check the proximity of the document consultation and ClickThrough Data uses machine learning that increases the complexity of the algorithm.

III. FILTERED-PAGE RANKING

In this section, we describe our proposed ranking method, called Filtered-Page Ranking (FPR). Before going into the details of the process, we first describe our notion of HTML page relevance and give a brief overview of the idea.

A. HTML page relevance

Some relevance criteria are based on a study of essential criteria for automatic indexing of text documents [26], where authors claim that to understand a document content the ideal is its full reading, although it is impractical. In that work, document segments and criteria that should be considered most important for indexing documents in digital media are defined. Considering some criteria defined in that work, in our proposal we believe that some criteria are more important to define the relevance of a HTML page: (i) the document title; (ii) the introduction and the first sentences of chapters/paragraphs; (iii) tables and lists; (iv) highlighted words; (v) the frequency of terms; (vi) stop words; and (vii) sentence-term. In addition to the criteria based on that study, since links are prominent elements of HTML pages, playing an important role in the design of web pages we also consider (viii) the number of links with all query terms used as a description of links. Intuitively, we can consider these components can represent very well a document without the need to consider the content as a whole.

As we are working with HTML document, we have made some adjustments in order to define the criteria: (i) title:

we consider the content in `title` and `meta` elements; (ii) `introduction and the first sentences of chapters and paragraphs`: our algorithm considers relevant the content that is close to the query terms in the document; (iii) `tables and lists`: all its contents is taken into account, being possibly represented, for example, by elements like `table`, `ul/ol`, `tr` and `li`; (iv) `highlighted terms`: they are emphasized in the text using specific HTML tags and can be underlined, bold or highlighted with different sizes or sources; these terms are taken into account on scoring an HTML document, increasing its relevance; (v) `frequency of terms`: the more a term of the user query appears in the document, the greater the relevance of the document.

Regarding `sentence-term`, the terms in a query tend to appear together. For example, when a user searches "recovery information", these two words tend not to be isolated (with no connection), they tend to appear near by, being terms of a sentence. The FPR penalizes web pages whose terms are far apart, as we can see in the correlation function in definition 6. `stop words` are irrelevant terms, without meaning that are not considered query keywords, usually represented by articles and prepositions.

B. Overview

The full process is executed over a DOM tree representation, which means the algorithm handles with nodes. There are two main and independent steps: (i) `page fragmentation and irrelevant content removal`: to eliminate those DOM nodes that have non-related information to the user query; and (ii) `document ranking`: to sort the relevant pages from a given query, making use of certain criteria indicating how close the document is to the query terms. The result generated from these steps is called `filtered DOM tree`.

For helping the process, the document metadata, containing information of the original document tree, are stored in the document repository. In general, the metadata comprise the document terms and their related nodes, as well as their properties (such as the HTML tag and the term occurrence in a node). Based on the terms used in the user query, the metadata presented on the filtered DOM tree are analyzed and used later to indicate the relevance of the this tree by calculating how close its content is to the query. Finally, the results are displayed in a ranking. If the filtered DOM tree does not have all mandatory terms, specified in the user query, it is not returned in the ranking. The way in which the metadata are stored in the repository depends on indexing methods and mapping structures, and it is not the focus of the work presented in this paper.

C. Query-Based Blocks Mining

The query-based blocks mining is the step of page segmentation and irrelevant content removal, in which the DOM tree is segmented into blocks. The blocks delimit the regions and the type of treatment performed over the DOM nodes. The objective of this phase is to extract a filtered tree that has only segmented blocks directly connected to the user query, discarding blocks with irrelevant contents.

1) Categorization of blocks: In this task, the DOM tree nodes are categorized into three groups: (i) `segmented blocks`; (ii) `disposal blocks`; and (iii) `highlighted blocks`. During the process, a categorized DOM tree is generated, whose categories are used to eliminate content, to extract useful content or to be used during the ranking phase.

Definition 1: `(Categorized DOM tree)`**:**

Let $N = \{n_1, ..., n_i\}$ be a set of nodes and $E = \{e_1, ..., e_i\}$ be the set of edges connecting the nodes in N. A categorized DOM tree DT is defined as a pair $DT = (N, E)$, where N is the set of nodes in which n_j is any node in A and can represent `segmented blocks`, `highlighted blocks` or `disposal blocks`.

A categorized DOM tree has both important nodes for the ranking process and nodes that must be eliminated. Those nodes can represent segmented blocks, highlighted blocks or disposal blocks, which can be treated as defined below.

Definition 2: `(Segmented Block)`**:** Let DT be a categorized DOM tree and n_j any node in DT. A block $Bsg = n_j$ is a sub-tree of DT called `segmented block`, such that n_j is any continuous region of the text, $Bsg \subset DT$.

Segmented blocks are sub-trees of the categorized DOM tree that are able to delimit regions, i.e., we consider segmented blocks to be elements that are capable of delimiting context (grouping HTML elements or sets of words that precede or follow the query keywords); a segmented block can be contained in others segmented blocks, generating nested segmented blocks. These regions may indicate blocks that contain the query terms and must be kept, as well as irrelevant content that must be eliminated. Generally, they delimit continuous regions of text or regions inserted within a context that groups them, such as, for example, tags `form` or `div`, which defines a set of data from a Web form or a given style and format, respectively. The HTML tags that can represent segmented blocks can be, for example, {`html, body, form, div, table, tr, iframe, article, section, ul, li, title, meta`}

Definition 3: `(Highlighted Block)`**:** Let n_j be any node in DT that can contain an HTML tag of character formatting. A block $Bhl = n_j$ is a sub-tree of DT called `highlighted block`, such that $Bhl \subset Bsg$ and Bhl is represented by a node that contains an element of character formatting.

Highlighted blocks are special blocks of a categorized DOM tree and are contained in a segmented block, representing HTML elements of character formatting. These elements format or highlight certain pieces of text, for example, they can underline text, mark bold or italic, and change the font size. A highlighted block is always contained in a segmented block and does not delimit regions considered text segment, it only highlights parts of continuous regions of text. It is not considered in the irrelevant content removal step, being preserved if its closest ascendant segmented block is also preserved. The highlighted blocks are important during the ranking step since they determine how close the text of the document content is to the query terms. They may be represented, for example, by the tags {`strong, b, i, u, span, a, h1, h2, h3, h4, h5, h6`}.

 ISBN: 978-1-61208-474-9

Definition 4: (Disposal Block): Let n_j be any node in DT that can contain an empty, invisible or hidden element. A block $Bdp = n_j$ is a sub-tree of DT, called disposal block, such that $Bdp \subset DT$ and Bdp is an empty (it does not contain text nor sub-trees) or invisible or hidden element.

Disposal blocks are automatically deleted since they represent the irrelevant content of the page and do not have visible text content. The entire sub-tree of a disposal block is deleted automatically when: (i) the node represents an empty element, i.e., it has no text itself; (ii) the node is a hidden or invisible element, not appearing in the presentation of the HTML page; containing, for example, attributes like "style" = "display: none", ("visibility" = "hidden" and "visibility" = "collapse".

2) Filtered tree generation: In the categorized DOM tree, the main node is the main segmented block, which may be composed of many others segmented blocks. The segmented blocks having user query terms compose the filtered DOM tree.

Definition 5: (Filtered DOM tree): Let DT be a categorized DOM tree, $C = \{t_1, ..., t_i\}$ a user query, $BDP = \{Bdp_1, ..., Bdp_m\}$ the set of all disposal blocks of DT, and $BSG_\phi = \{Bsg_1, ..., Bsg_n\}$ the set of segmented block of DT such that $BSG_\phi \not\subset C$. A filtered DOM tree Af is a tree such that $Af = DT - BSG_\phi - BDP$.

A filtered DOM tree consists only of segmented blocks that contain the user query terms, without the disposal blocks. The segmented blocks that do not have any of the query terms are discarded. In nested segmented blocks, the children blocks that do not have any query terms are excluded, preserving the ascendant segmented blocks if it, or at least one child segmented block, has at least one query term.

D. The ranking function

Before to introduce the ranking function, it is important to define the terms correlation function. For classic information retrieval models, the terms in a document are assumed to be mutually independent, which means a given term t_i tells us nothing about t_{i+1}. However, the terms occurrences are not uncorrelated. For example, the terms 'information' and 'retrieval' tend to appear together in a document about information retrieval systems [27]. In that document, the appearance of one of these terms attracts the appearance of the other. Thus, they are correlated and we must reflect this correlation. In this paper, this correlation is measured by means of the distance between terms, according to Definition 6 and Equation 1.

Definition 6: (Correlation Function): Let $C = \{t_1, ..., t_n\}$ be a query and DT a categorized DOM tree. The correlation between terms in C and terms in DT is measured by the following function:

$$D(C, DT) = \begin{cases} 1, & \text{if } d(t_i, t_j) < th \\ \alpha, & \text{otherwise} \end{cases} \quad (1)$$

where th is the threshold that indicates a minimum distance between terms, inside the categorized DOM tree, and α is a value in the interval [0..1] used to penalize a given page when the distance between terms is bigger than th. The distance function $d(t_i, t_j)$ assigns a character distance value to each pair of term t_i and t_j (this distance can be calculated by any character distance function [28]).

In order to be in a top position in the ranking, the DOM tree has to have a minimum content related to the query, which can be in the text flow or in the links (typical case of e-commerce pages). This intuition is computed as defined in Equation 2.

Definition 7: (Page-relevance Function): Let $C = \{t_1, ..., t_n\}$ be a query, DT a categorized DOM tree, $f_t t(DT)$ a function that returns the total number of terms in DT and $L = \{l_1, ..., l_k\}$ the set of k links in DT that have all terms of C. The Page-relevance is given by the following function:

$$CP(C, DT) = \begin{cases} 1 & \text{if } f_{tt}(DT) > x \vee k > y \\ \beta & \text{otherwise} \end{cases} \quad (2)$$

where x indicates the minimum amount of terms a page must have, y represents the minimum amount of links with all query terms the page must have and β is a value in the interval [0...1] used to penalize the page position.

The ranking function is defined taking into account the relevance criteria described in Section III-A, considering the importance of certain parts of the document (title, tables, highlights, for example), and the number of occurrences of query terms in certain parts of the document.

Definition 8: (Ranking function): Let $C = \{t_1, ..., t_n\}$ be a query, DT a categorized DOM tree and $L = \{l_1, ..., l_k\}$ the set of k links in DT that have all terms of C. The ranking function $R(C, DT)$, which returns a score between C and DT, is:

$$\begin{aligned} R(C, DT) =& D(C, DT).CP(C, DT).(W_1.\sum_{i=1}^{n}(f_o(t_i, DT))) + \\ & (W_2.\sum_{i=1}^{n}(f_{hb}(t_i, DT))) + (W_3.k) + \\ & (W_4.\sum_{i=1}^{n}(f_{tm}(t_i, DT))) + (W_5.f_t(DT)) \end{aligned} \quad (3)$$

where
$D(C, DT)$ is the correlation function;
$CP(C, DT)$ is the page-relevance function;
$f_o(t_i, DT)$ is a function that returns the number of occurrences of a term t_i in DT;
$f_{hb}(t_i, DT)$ is a function that returns the number of occurrences of the term t_i in highlighted blocks of DT;
$f_{tm}(t_i, DT)$ is a function that returns the number of occurrences of a term t_i in the main title or in metadata of DT;
$f_t(DT)$ is a function that returns the total terms of DT;
W_i: the weight of each criterion.

The intention behind the ranking function $R(C, DT)$ is to calculate the proximity of a categorized DOM tree A with terms in C, using the relevance criteria presented in Section III-A, given a weight to each one. Furthermore, those pages, in which the distance between query terms are bigger than a threshold, or that do not have a minimum content related to the query, are penalized, respectively by means of the functions $D(C, DT)$ and $CP(C, DT)$.

font=footnotesize,sc,justification=centering,labelsep=period

TABLE II. RECALL X QUERY-BASED BLOCKS MINING AND CETR PRECISION.

Page	Total of Terms	tRel-A	t-Af		tRel-Afilt		Recall		Precision		F-Value	
			QBM	CETR	QBM	CETR	QBM	CETR	QBM	CETR	QBM	CETR
1	2223	1964	1759	1631	1759	1538	0.896	0.783	1.000	0.943	0.948	0.856
2	618	163	464	442	160	145	0.982	0.890	0.345	0.328	0.510	0.479
3	1078	738	811	28	733	0	0.993	0	0.904	0	0.946	0
4	5879	5108	3339	1912	3291	1841	0.644	0.360	0.986	0.963	0.815	0.525
5	2816	1855	1826	1836	1793	1821	0.967	0.982	0.982	0.992	0.975	0.987
6	623	328	328	322	328	322	1.000	0.982	1.000	1.000	1.000	0.991
7	1207	389	288	703	288	348	0.740	0.895	1.000	0.495	0.87	0.637
8	1311	0	868	1023	0	0	0	0	0	0	0	0
9	703	348	314	374	293	328	0.842	0.943	0.933	0.877	0.885	0.909
10	1722	1308	1271	1229	1270	1189	0.971	0.909	0.999	0.967	0.985	0.937
-	-	-	-	-	Average		0.803	0.674	0.815	0.657	0.787	0.632

IV. EXPERIMENTAL EVALUATION

In this section, we describe the experiments we performed to demonstrate the effectiveness of our proposal. The experiments have the following main goals: (i) to analyse the query-based blocks mining, aiming at evaluating its effectiveness in segmenting an HTML page and removing irrelevant content from the page; (ii) to perform a comparative analysis among different combination of removal algorithm and ranking algorithm; and (iii) analyse the FPR process itself.

A. Methodology and Evaluation Metrics

The total set of documents used in the experiments consists of 1,530 Web pages, collected from different news and entertainment websites. The queries were associated with five different domains: history, law, diseases, electronics and politicians. The different domains have been chosen in order to identify if any of them would behave differently from others. As our ranking function uses different weights, we have set them as follow. The number of occurrences of a term t_i in A: $W_1 = 9.98$; the number of occurrences of the term t_i in highlighted blocks of A: $W_2 = 15$; the number of links in A that have all terms of C: $W_3 = 15$; the number of occurrences of a term t_i in the main title or in metadata of A: $W_4 = 60$; the total terms of A: $W_5 = 0.02$. The values used to penalize the page position: α = 0.08, β= 0.1.

The weights were manually calibrated based on observations of the database metadata. For the manual calibration, the weights were given initial values and adjusted for more or for less to best suit the improvement of the precision and recall of FPR ranking under original web pages (without filter). It is common to find Web pages with more than 10,000 or 20,000 words. Therefore, an apparently unimpressive weight of 0.02 found for the number of words becomes as significant as the other criteria used in the final ranking process.
Web pages that satisfy 30 queries in these 5 different areas were collected from google and classified by 5 different users to determine their relevance. Each page were scored from 1 to 4 in the following scale: insignificant (1), low significance (2), significant (3) and very significant (4). The pages with an average score higher than or equal to 3 were classified as being 'relevant' and the pages with an average score lower than 3 were classified as being 'irrelevant'. For each query the number of irrelevant pages is greater than or equal to the number of relevant pages and there are at least 10 relevant pages for each query.

Lucene was used as the baseline, because it is widely used in tools for local search with implementation (VSM) available and it is based on the performed query like FPR. It does not have the limitations as requiring recognition of users (ClickThrough Data), the use of A.I (PCR, ClickThrough Data) and the needs that many pages share the same template (Computing Block Importance for Searching on Web Sites).
Block-Based Web Search have improvements compared to FixedPS and uses Web Pages. In section V it is mentioned that comparisons can be made between FPR/QBM and Block-Based Web Search, with the improvement of collecting the text content from tag <body> instead of <title> on method Block-Based Web Search.
As the baseline irrelevant content removal algorithm, we choose the CETR algorithm. The tests have been conducted with the following configurations: (i) Lucene: ranking algorithm of the classic vector model; (ii) FPR: our proposed ranking algorithm; (iii) FPR + CETR: our proposed ranking algorithm, on the basis of filtered documents through CETR algorithm; and (iv) FPR + QBM: our proposed ranking algorithm, on the basis of filtered documents through our irrelevant content removal algorithm.

The metrics we have used for evaluation were that from classical information retrieval community [27]: recall, precision and F-measure. As usual, the recall value was obtained by the ratio of relevant documents by each query, which in fact were recovered. The precision was calculated by the proportion of recovered material that were relevant, and F-measure is the harmonic mean of recall and precision.

font=footnotesize,sc,justification=centering,labelsep=period

TABLE III. PRECISION.

Ranking	P@10	P@15	P@20	P@10	P@15	P@20
	History			Law		
FPR+QBM	0.76	0.64	0.53	0.78	0.62	0.49
FPR(-)	0.78	0.60	0.49	0.73	0.55	0.40
FPR+CETR	0.64	0.53	0.45	0.70	0.617	0.51
Lucene	0.46	0.45	0.44	0.55	0.55	0.51
	Diseases			Electronics		
FPR+QBM	0.85	0.77	0.66	0.83	0.76	0.70
FPR(-)	0.82	0.72	0.59	0.70	0.67	0.59
FPR+CETR	0.78	0.67	0.58	0.73	0.62	0.53
Lucene	0.70	0.63	0.61	0.60	0.58	0.55
	Politicians			All		
FPR+QBM	0.87	0.69	0.53	0.81	0.69	0.58
FPR(-)	0.80	0.64	0.53	0.77	0.63	0.52
FPR+CETR	0.77	0.64	0.50	0.72	0.61	0.51
Lucene	0.83	0.644	0.54	0.61	0.56	0.53

B. Results

We now describe the experiments used to evaluate our proposed algorithms. We first present the QBM effectiveness in eliminating irrelevant content, and then provide a comprehensive evaluation of the ranking proposal.

 ISBN: 978-1-61208-474-9

1) Analysis of QBM: The QBM algorithm was analyzed in order to evaluate its effectiveness in removing irrelevant content from HTML pages, comparing it with a baseline, the CERT algorithm [16]. For this purpose, the following evaluation parameters were considered: (i) `tRel-Af`: total of relevant terms of the filtered DOM; (ii) `t-Af`: total of terms of the filtered DOM; (iii) `tRel-DT`: total of relevant terms of the original DOM. Using these parameters, it was possible to assess the precision and the recall as follows: $recall = (tRel-Af)/(tRel-DT)$; $precision = (tRel-Af)/(t-Af)$.

In general, the QBM results reached 80% to 85% of precision, being able to eliminate almost all the irrelevant content of the pages in many cases. As we can observe in Table II, our proposal has surpassed the baseline. The page listed as number 8 had 0% of precision and recall. This happens due to the fact that QBM and CETR does not consider semantic. For example, in a query "ceara history", in which the user's interest is related to the Ceara State history, the query can match it with a page of the history of Ceara Sporting Club. The same happens with the page indicated by number 2, which does not match the query "public service definition" with a relevant page because it brings a page about the definition of "public servant", in which, within the same segmented block, there is a text about "public service definition", i.e., only a small part of the document relates directly to the subject "public service definition". In page 3, CETR extracts the document main region, but having only irrelevant nodes to the query.

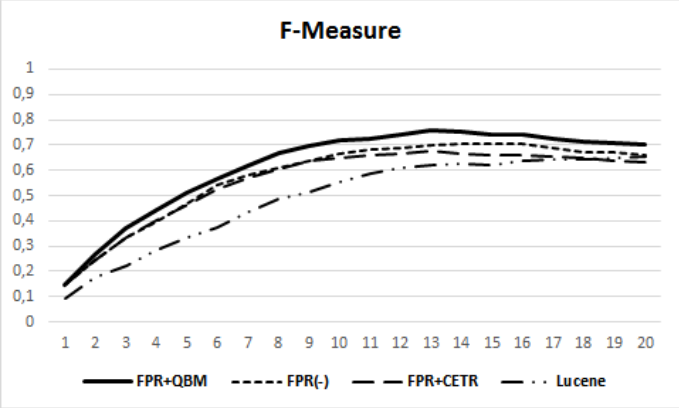

Figure 1. Results with f-measure.

2) Comparative Analysis: The results obtained in the comparative analysis were performed in two ways: (i) each domain result was individually analyzed in order to identify if any of them would behave differently from the general rule; and (ii) the overall results was analyzed, considering average values over the entire set of documents (1.530 documents), independent of domain, in order to obtain a general idea of its behavior.

In Table III, we present the precision results from experiments on three different rankings: P@10, P@15 and P@20. Analyzing the table, we can see that in the first 10 positions the combination of our two proposals, FPR+QBM, has the best ranking. This shows that our FPR ranking algorithm works well when used together with a good irrelevant content removal algorithm.

The results of the F-Measure values evaluation in the first 20 positions are shown in Figure 1. Figure 2 shows the curves of recall/precision values. Considering that FPR+QBM has, in the first twenty positions, average values of Precision and F-Measure better than Lucene, FPR+CETR and FPR (without filter), the effectiveness of our proposal is reached.

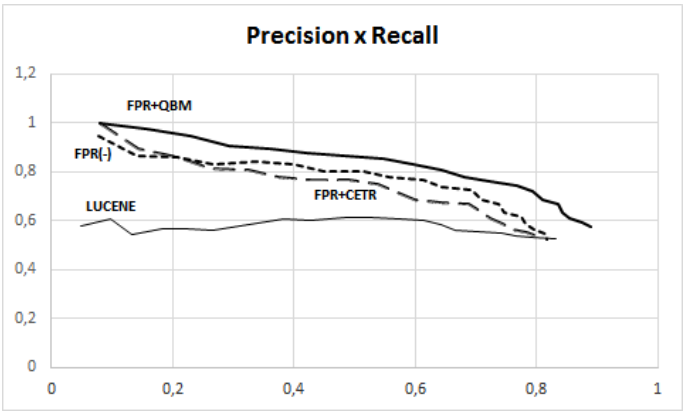

Figure 2. Precision x recall curve.

Analyzing the results, it is clear that the average recall, precision and F-measure on the first 10 positions are higher with the application of the proposed method than with the use of Lucene, which uses the vector model to define how close a document is to the query.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a filtered-page ranking process based on the user query terms, relevance criteria involving the importance of certain parts of the document and highlighted aspects of certain components. The process involves segmentation of HTML pages and irrelevant content removal. The documents are segmented into blocks and those considered as irrelevant are deleted. Our proposed ranking method called Filtered-Page Ranking (FPR) works with prior elimination of irrelevant content, which is a satisfactory process when compared to some literature methods, and that can be used to define the relevant HTML pages in relation to a given query. As future work, we intend to find an optimum weight method of the important criteria for defining the ranking, and provide new relevant criteria for defining ranking and compare FPR/QBM with anothers methods specific of Web Pages like Block-Based Web Search (with the improvement of collecting the text content from tag `body` instead of tag `title`).

This paper presents a filtered-page ranking process based on the user query terms, relevance criteria involving the importance of certain parts of the document and highlighted aspects of certain components. The process involves segmentation of HTML pages and removal irrelevant content. The documents are segmented into blocks and those considered as irrelevant are deleted. Our proposed ranking method called Filtered-Page Ranking (FPR) works with prior elimination of irrelevant content, which is a satisfactory process when compared to some literature methods, and that can be used to define the relevant HTML pages in relation to a given query. As future work, we intend to find an optimum weight method of the important criteria for defining the ranking, and provide new relevant criteria for defining ranking and compare with other methods specific of Webpages like Block-Based Web Search

 ISBN: 978-1-61208-474-9

(with the improvement of collecting the text content from tag body instead of tag title).

REFERENCES

[1] A. N. Langville and C. D. Meye, Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton Ed., 2011.

[2] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine." Computer Networks, vol. 56, no. 18, 2012, pp. 3825–3833.

[3] A.-J. Su, Y. C. Hu, A. Kuzmanovic, and C.-K. Koh, "How to improve your search engine ranking: Myths and reality," ACM Trans. Web, vol. 8, no. 2, Mar. 2014, pp. 8:1–8:25.

[4] A. Karatzoglou, L. Baltrunas, and Y. Shi, "Learning to rank for recommender systems," in Proc. 7th ACM RecSys, 2013, pp. 493–494.

[5] L. Lerche and D. Jannach, "Using graded implicit feedback for bayesian personalized ranking," in Proc. 8th ACM RecSys, 2014, pp. 353–356.

[6] Y. Song, L. Zhang, and C. L. Giles, "Automatic tag recommendation algorithms for social recommender systems," ACM Trans. Web, vol. 5, no. 1, Feb. 2011, pp. 4:1–4:31.

[7] K. Balog and H. Ramampiaro, "Cumulative citation recommendation: Classification vs. ranking," in Proc. 36th ACM SIGIR, 2013, pp. 941–944.

[8] G. Berardi, A. Esuli, and F. Sebastiani, "Utility-theoretic ranking for semiautomated text classification," ACM Trans. Knowl. Discov. Data, vol. 10, no. 1, Jul. 2015, pp. 6:1–6:32.

[9] J. Fang, L. Guo, X. Wang, and N. Yang, "Ontology-based automatic classification and ranking for web documents," in Proc. 4th IEEE FSKD. IEEE Computer Society, 2007, pp. 627–631.

[10] J. Li, B. Saha, and A. Deshpande, "A unified approach to ranking in probabilistic databases," The VLDB Journal, vol. 20, no. 2, Apr. 2011, pp. 249–275.

[11] Y. Chen, X. Li, A. Dick, and R. Hill, "Ranking consistency for image matching and object retrieval," Pattern Recogn., vol. 47, no. 3, Mar. 2014, pp. 1349–1360.

[12] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Ranking fraud detection for mobile apps: A holistic view," in Proc. 22nd ACM CIKM, 2013, pp. 619–628.

[13] M. P. Selvan, A. C. Sekar, and A. P. Dharshini, "Survey on web page ranking algorithms," International Journal of Computer Applications, vol. 41, no. 19, 2012, pp. 1–7.

[14] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manage., vol. 24, no. 5, Aug. 1988, pp. 513–523.

[15] "Lucene," https://lucene.apache.org/core, accessed: 2016-04-12.

[16] T. Weninger, W. H. Hsu, and J. Han, "Cetr: content extraction via tag ratios," in Proc. 19th WWW. ACM, 2010, pp. 971–980.

[17] "The pagerank citation ranking: bringing order to the web," http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf, accessed: 2016-04-11.

[18] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," Journal of the ACM (JACM), vol. 46, no. 5, 1999, pp. 604–632.

[19] N. Duhan, A. Sharma, and K. K. Bhatia, "Page ranking algorithms: a survey," in IEEE IACC, 2009, pp. 1530–1537.

[20] T. Joachims, "Optimizing search engines using clickthrough data," in Proc. 8th ACM SIGKDD, 2002, pp. 133–142.

[21] J. Pokorny and J. Smizansky, "Page content rank: an approach to the web content mining," in Proc. IADIS Conf. On Applied Computing, vol. 1, 2005, pp. 289–296.

[22] E. Hatcher, O. Gospodnetic, and M. McCandless, Lucene in action. Manning Publications Greenwich, CT, 2004.

[23] J. P. Callan, "Passage-level evidence in document retrieval," in Proc. of the 17th ACM SIGIR conference on Research and development in information retrieval, 1994, pp. 302–310.

[24] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Block-based web search," in Proc. 27th ACM SIGIR, 2004, pp. 456–463. [Online]. Available: http://doi.acm.org/10.1145/1008992.1009070

[25] D. Fernandes, E. S. de Moura, B. Ribeiro-Neto, A. S. da Silva, and M. A. Gonçalves, "Computing block importance for searching on web sites," in Proc. 16th ACM CIKM, 2007, pp. 165–174.

[26] G. Borges, G.; Lima, "Automatic indexing of text documents: Essential criteria proposal," Journal of Information Research, vol. 3, no. 1, 2014, pp. 360–370.

[27] R. A. Baeza-Yates and B. A. Ribeiro-Neto, Modern Information Retrieval. ACM Press / Addison-Wesley, 1999.

[28] "Simetrics," http://sourceforge.net/projects/simmetrics/, accessed: 2016-04-11.

 ISBN: 978-1-61208-474-9