ISSN No. 0976-5697

Volume 5, No. 7, September-October 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Performance Metrics for Selection of Quality Hidden Web Documents

Rashmi Agarwal
Lecturer
JP Institute of Engineering and Technology,
Meerut, India

Niraj Singhal
Assistant Professor
JP Institute of Engineering and Technology,
Meerut, India

Prem Sagar
Associate Professor
Shobhit University, Meerut, India

Abstract: Hidden web continues to grow as organizations with large amount of high-quality information are placing their content online, providing web-accessible search facilities over existing databases. In particular, some extravagant web pages containing query search form, redundant data also retrieved while extracting content from hidden web. This paper addresses the issues related to selecting hidden web documents. It introduces a generic operational model for selection of quality hidden web documents. It also describes how this model helps in extracting quality hidden web documents and ignoring web pages which do not include form, downloads non-query forms and remove all the redundant query form within the same domain.

Keywords: Hidden web, quality documents, performance metrics, search forms, non-query form, submission efficiency, cost analysis.

I. INTRODAUCTION

The visible web is what one can find using traditional web search engines while invisible web (or deep web) contains information that isn't indexed by traditional search engines. Recent studies show that a significant fraction of web is hidden behind search form, in large searchable electronic databases. Pages in the hidden web are dynamically generated in response to queries submitted via the search forms. The hidden web continues to grow, as organizations with large amounts of high-quality information (patents and trademarks office, news media companies etc.) are placing their content online, providing web-accessible search facilities over existing databases. Approximately 100,000 hidden web sites currently exist on the web [2,3].

Some contents that remains hidden from general search engines are [1]:-

(i) The contents of searchable databases: When user searches in a library catalog, article database, statistical database, etc., the results are generated "on the fly" in answer to his search. Because the crawler programs cannot type or think, they cannot enter passwords on a login screen or keywords in a search box. Thus, these databases must be searched separately.

(ii) Excluded pages: Search engine companies exclude some types of pages by policy, to avoid cluttering their databases with unwanted content.

(iii) Dynamically generated pages of little value beyond single use: Consider billions of possible web pages generated by searches for books in library catalogs, public-record databases, etc. Each of these is created in response to a specific need. Search engines do not want all these pages in their web databases, since they generally are not of broad interest.

(iv) Pages deliberately excluded by their owners: A web page creator who does not want his/her page showing up in search engines can insert special "meta tags" that will not display on the screen, but will cause most search engines' crawlers to avoid the page.

The deep web holds academic studies and papers, scientific research, government publications, electronic books, bulletin boards, mailing lists, online card catalogs, articles, directories, many subscription journals, archived videos, images and more. Building a hidden-web crawler that can automatically download pages from the hidden web, so that search engines can index them is not an easy task.

This paper presents performance metrics architecture for selection of quality hidden web contents. It identifies web page templates and the tag structures of a document, in order to extract structured data from hidden web sources as the results returned in response to a user query are typically presented using template generated web pages.

II. RELATED WORK

The fundamental difference between the performance metrics of a hidden web crawler and that of a traditional crawler [4, 5] is with respect to pages containing search forms. The crawler module retrieves pages from the web for later analysis by the indexing module. A crawler module typically starts with an initial set of URLs, say S_0.Roughly, it places S_0 in a queue, where all URLs to be retrieved are kept and prioritized. From this queue, the crawler gets a URL (in some order), downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated until the crawler decides to stop [8].

A. Submission Effienciency:

Raghavan et al approach [5] considers a coverage metric that measures the ratio of the number of 'relevant'

pages extracted by a crawler to the total number of 'relevant' pages present in the targeted hidden databases. Let N_{total} be the total number of forms that the crawler submits, during the course of its crawling activity. Let $N_{success}$ denote the number of submissions which result in a response page containing one or more search results. Then, the strict submission efficiency (SE_{strict}) metric is defined as:-

$$SE_{strict} = \frac{N_{sucess}}{N_{total}} \quad \text{-(1)}$$

They also define a lenient submission efficiency ($SE_{lenient}$) metric that penalizes a crawler only if a form submission is semantically incorrect (e.g., submitting a company name as input to a form element that was intended to receive names of company employees). Specifically, if N_{valid} denotes the number of semantically correct form submissions, then

$$SE_{lenientt} = \frac{N_{valid}}{N_{total}} \quad \text{-(2)}$$

$SE_{lenient}$ is more difficult to evaluate, since each form submission must be compared manually with the actual form, to decide whether it is a semantically correct. For large experiments involving hundreds of form submissions, computing $SE_{lenient}$ becomes highly cumbersome.Ntoulas et al [4] formalized the problem of query selection. They assume that the crawler downloads pages from a web site that has a set of pages 'S' (the rectangle shown in Figure 1). Each web page in 'S' is represented as a point. Every potential query qi that a user may issue can be viewed as a subset of 'S', containing all the points (pages) that are returned when user issues qi to the site. Each subset is associated with a weight that represents the cost of issuing the query. Under this formalization, their goal is to find which subsets (queries) cover the maximum number of points (web pages) with the minimum total weight (cost). T

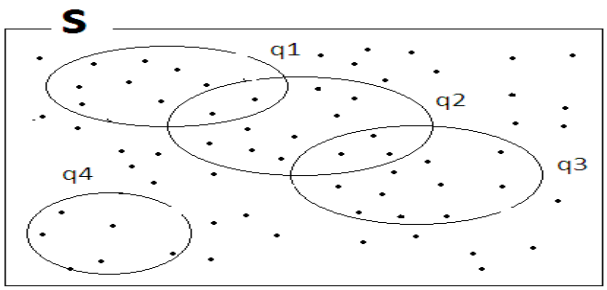

Figure 1. A set-formalization of the optimal query selection problem

There are two main difficulties that need to be addressed in this formalization. First, in a practical situation, the crawler does not know which web pages will be returned by which queries, so the subsets of 'S' are not known in advance. Without knowing these subsets the crawler cannot decide which queries to pick to maximize the coverage. Second, the set-covering problem is known to be NP-Hard [4], so an efficient algorithm to solve this problem optimally in polynomial time has yet to be found.

Ipeirotis and Gravano [12] gives a document sampling technique for text databases that results in higher quality database content summaries than those by the best known algorithm. The aim of form analysis is to process a form page and extract all the information necessary to build the internal representation of the form. For Hi WE, the main challenge in form analysis is the accurate extraction of the labels and domains of form elements. For accessing the "server-side" deep web, Álvarez [14] gives the idea of deepBot which can be provided with a set of domain definitions, each one describing a certain data-gathering task. DeepBot automatically detects forms relevant to the defined tasks and executes a set of predefined queries on them.

B. Page Rank Updating:

Page Rank [7,9] is a numeric value that represents how important a page is on the web. Adamic and Huberman [13] said that web site growth and popularity actually follow rules which are useful for predicting the web's future behavior**.** Page Rank[6] is the Google's method of measuring a page's "importance". Google uses the Page rank to adjust result so that more important pages moves up in the results page of user's search result display. It will update the rank of page after searching the page.

$$PR(u) = (1-d) + d(PR(V_1)/N(V_1)+....PR(V_N)/N(V_N)) \quad \text{-(3)}$$

III. PROPOSED WORK

While doing query based searching, the search engines return a list of web documents containing both relevant and irrelevant pages and sometimes show the higher ranking to the irrelevant pages as compared to relevant pages [10]. The technique of hidden page selection can be formalized as follows.

Let 'SOP' be the set of hidden web pages that the crawler downloads from the website, in which each web page is represent as a point (dots as shown in figure 2) and each web pages is returned when a query issue to the site. And out of these pages which contain all these attributes are selected as quality hidden web documents. As shown in figure 2 the intersection of p1, p2, p3 (black surface containing dots) results quality hidden web documents which contain all these attributes within each subset (as shown in figure 2), one can extract :-

Attribute 1:-Web pages containing only form.

Attribute 2:-Only query forms are included (excludes non-query form).

Attribute 3:-Remove all redundant query form within the same domain.

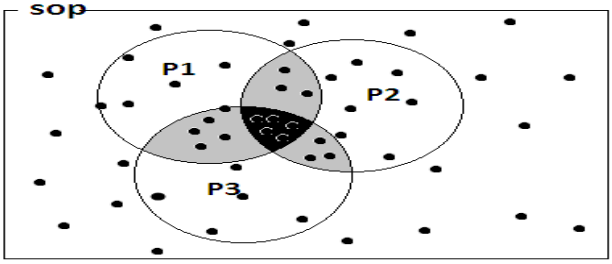

Figure 2. A set formalization of optimal page selection technique

Their algorithm leverages the observation that although one may do not know which pages will be returned for a particular query 'Q' that is issued, one can predict how many pages will be returned. Based on this information our page selection algorithm can then select the "best" web pages that cover the content of the web site as well as which is more require by each and every user while issuing the query.

A. Performance metrics:

Suppose a user uses a search form to submit queries on a hidden database (as shown in figure 3) then after filling a form, a crawler receives it which contain four components i.e. Internal Form Representation, Task-specific database, Matching function and response analysis**.** At first, web crawler builds an internal representation of a form 'f' which contain a set of 'n' form elements, submission information associated with the form and meta-information about the form. Task-specific database helps in containing necessary information to formulate queries relevant to the particular task. Matching function helps in associating value with each element and finally, Response analysis stores the response page in the repository. Then, out of those web pages, pages containing only form including query form which does not contain redundant query form within the same domain are extracted from hidden database.

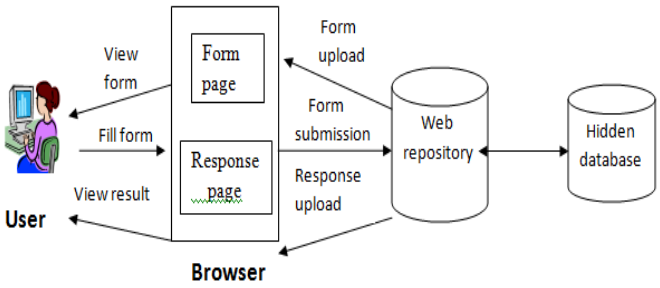

Figure 3. User form interaction

The process by which web form repository work is discuss below:-

Algorithm(extract web pages from web repository)

i) Extract all web pages(WP)
ii) Extract only those web pages which includes form from WP
Wf_i = select_form(WP)
iii) Remove all non-query form from Wf_i
QF_i = Query_form(Wf_i)
iv) Remove all redundant query form
Q^f= Remove_redundancy(QF_i)
v) Download Q^f
vi) done

Figure 4 represent the internal representation of crawler as well as hidden database to represent how to extract quality hidden web documents i.e., out of hidden web pages, pages containing only form as well as including query form which does not contain redundant query form within the same domain are extracted from hidden database. As a result, response page is uploaded to provide response to the query issued by the user.

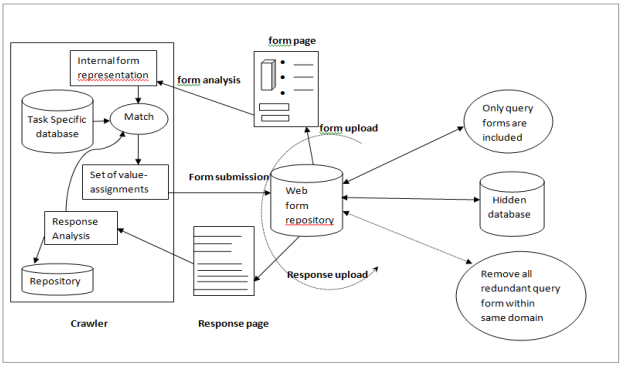

Figure 4. Crawler form interaction

In order to check whether a page pi contain all the three attributes, one may take help of the following charts to know pi in SOP contain all three attributes of which range or quantity.

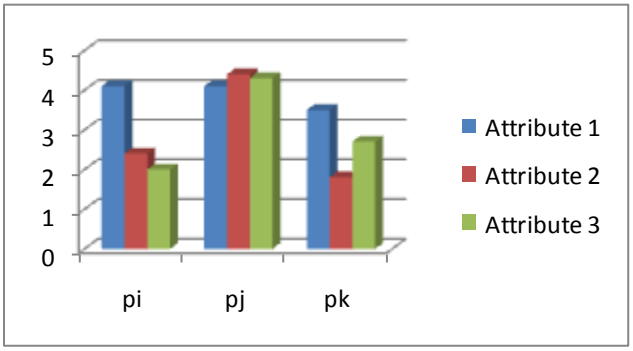

Figure 5. Graph to represent the proportion of all three attributes contain by each page pi, pj and pk

In the most common case, the page selection cost consists of a number of factors, including only form, exclude non-query form and all redundant query form. Search engines use web crawlers to collect these documents from web for the purpose of storage and indexing. An incremental crawler visits the web for updating its collection. There is a need to regulate the frequency of the crawler to visit web sites and provide latest information to the user [9]. Given a query qi, we use P(qi) to denote the fraction of pages that we will get back ,if one issue query qi to the site. In this paper, an search technique is used to represent the quality hidden web pages which is described by means of algorithm in which the set of pages p1,p2,p3 is retrieved while issuing a particular query 'Q'. Then, on the basis of union and intersection of pages, cost and probalbility of selected quality hidden web pages is estimated. Then, on the basis of cost and probability, efficiency is calculated and finally rank of the hidden web page[6] is determined whose procedure in the form of algorithm is shown below :-

Algorithm :- (*Extract Quality Hidden Web Page*)

a. When we issue a query to the site, fraction of pages will returned for this :-

$Q(p1 \cup p2 \cup p3)$is used to represent the fraction of pages that are returned from either page p1 or page p2 or page p3 (i.e. the union of Q(p1) and Q(p2) and Q(p3)).

(i) $Q(p1 \cap p2 \cap p3)$ is used to represent the fraction of pages that are returned from both p1 and p2 and p3(i.e. the intersection of Q(p1) and Q(p2) and Q(p3)).

b. Selection of Cost :-

Then, we calculate cost(pq) to represent the cost of retrieving web page from particular query

$$\mathbf{Cost(p^Q) = C_p + Cr\ (q) + C_d} \qquad \textbf{-(4)}$$

Where C_p is fixed cost of retrieving web form.

Cr(q) is cost for downloading the response page which includes only query form.

C_d is cost for downloading the matching pages which does not includes redundancy of query form.

c. Estimating probability of page containing all three attributes:-

In order to identify the pages which is more desirable. At first, we need to estimate whether the page contain all

three attributes. Let A1 be an attribute for checking whether the web page contains only forms, A2 be an attribute for checking only query form and A3 be an attribute for checking the redundancy in the query form. So, A3 be calculated as:

$$A3 = (p_{10} \cup p_{11} \cup p_{12} \cup ..) \cap (p_{20} \cup p_{21} \cup p_{22} \cup ..) \cap (p_{30} \cup p_{31} \cup p_{32} \cup ..) \quad -(5)$$

Where p_{10},p_{11},p_{22} be the pages of set p1 while p_{20},$_{21}$,p_{22}…be the pages of set p2 and p_{30},p_{31},p_{32} be the pages of set p3

Then, probability of page p_i from the set of pages p1,p2,p3 containing all three attribute is :-

$$\textbf{Probability}(p^Q) = \frac{p((p_i \cap A1) \cup (p_i \cap A2) \cup (p_i \cap A3))}{(A1 \cup A2 \cup A3)} \quad \textbf{-(5)}$$

d. Determining efficiency of page :-

There are two factors that taken into account, the no. of pages containing all three attributes and cost of retrieving these web pages while issuing a particular query for example:- if two pages p_i and p_j incurs the same cost but p_i matched to more common attributes than p_j, p_i is more desirable than p_j. Similarly, if p_i and p_j both contain all these attributes, but p_i incurs less cost than p_j, p_i is more desirable. Based on this observation, the following efficiency metrics is:-

Algorithm:- Greedy Select Term()

Parameters:-

Q:- Q is a particular query issued by the user

P^Q:- P^Q is a page selected as quality hidden web page

SOP:- SOP is a set of pages of p1,p2,p3 for a particular query

Procedure:-

(i) For each p^Q in SOP do

(ii) Estimate Efficiency(Ei) = $\frac{p^Q(Q)}{cost(p^Q)}$

(iii) Done

(iv) Return p^Q of maximum efficiency

To quantify desirability of pages from the query ‘Q’

$$\textbf{Efficiency(Ei)} = \frac{p^Q(Q)}{cost(p^Q)} \quad \textbf{-(6)}$$

Based on the value of cost and probability of selected quality hidden web page efficiency(Ei) of that page is estimated, by which the rank of the quality hidden web page is determined. The graph shown below is used to predict the value of cost, probability and efficiency of quality hidden web pages and on the basis of these value, rank of the quality hidden web page is retrieved for a particular query ‘Q’.

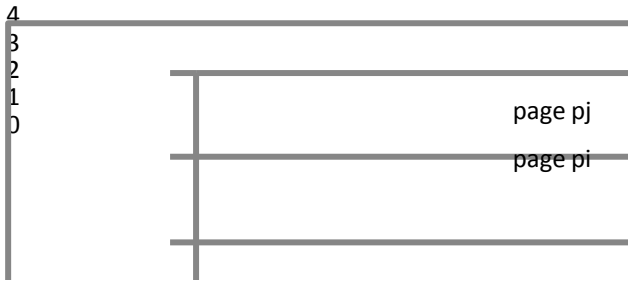

Figure 6. Graph to represent the value of cost, probability, efficiency and rank of any two hidden web page selected as quality hidden web page

B. Computation of page selection statistics:

The main idea for the page selection statistics table is to select only those hidden web pages that contain only form, and non-query forms are not included as well as all redundant query form within the same domain were removed. These matched pages are recorded into the table. For example, on fetching the query ‘qi’ “shobhit university”, then out of those web pages which are retrieved, select only those web pages which have three common attributes i.e web pages that contain only search form, web pages which include only query form (exclude non-query), and all redundant query form within same domain are removed.

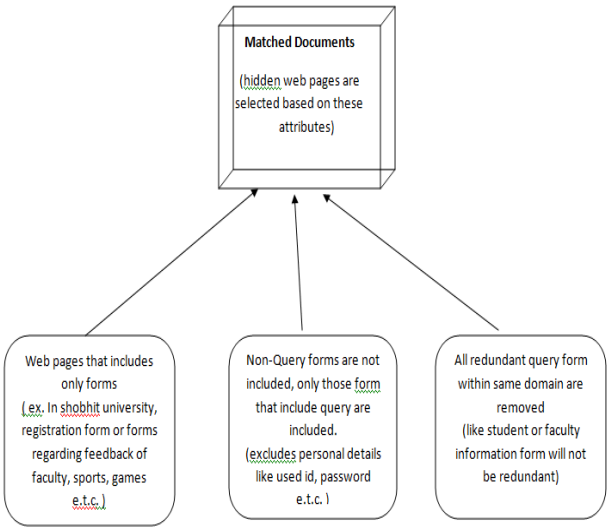

Figure 7. Updating the page selection statistics table

IV. CONCLUSION

An operational model for searching hidden web documents is presented that describes the steps that web repositary must take, in order to finding quality hidden web documents and on the basis of the quality hidden web pages which are returned, performance metrics are also introduced which calculate the cost of page selection and its efficiency while issuing query to the site.

V. FUTURE SCOPE

As discussed above Algorithms are tested to extract hidden web documents. In order to get relevant hidden web documents, various formula’s are used to calculate cost, probability and its efficiency of hidden web documents but selection of quality data from the hidden web is not an easy task so it need further researches to carry this task. In future some other approaches can be applied to calculated rank of hidden web documents that would increase the chance to get more relevant hidden web documents.

VI. REFERENCES

[1]. http://www.lib.berkeley.edu

[2]. http://www.completeplanet.com.

[3]. http://websearch.about.com/od/invisibleweb/tp/deep-web-search-engines.htm

[4]. Alexandros Ntoulas, Petros Zerfos and Junghoo Cho, “Downloading Textual Hidden Web Content”, Proceedings of 5th ACM/IEEE-CS Joint Conference on Digital libraries, New York, USA, pp. 100-109, 2005.

[5]. Sriram Raghavan and Hector Garcia-Molina, "Crawling on Hidden Web", Proceedings of 27th International Conference on Very Large Databases , USA, pp.129-138, 2001.

[6]. George Valkanas, Alexandros Ntoulas and Dimitrios Gunopulos, "Rank-Aware Crawling of Hidden Web sites", Proceedings of 14th International Workshop on the Web and Databases, Athens, Greece, 2011.

[7]. Priyanka Jain and Megha Bansal, "Efficient Crawling the Deep Web", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue. 5, pp. 620-623, 2014

[8]. Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke and Sriram Raghavan, "Searching the web", Journal of ACM Transactions on Internet Technology, Vol. 1, Issue. 1, pp. 2-43, 2001.

[9]. Arun Kumar Singh and Niraj Singhal, "A Novel Page Rank Algorithm for Web Mining based on User's Interest", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, Issue. 9, pp. 395-400, 2012.

[10]. Sunil Kumar and Niraj Singhal, "Ignoring Irrelevant Pages in Weighted Page Rank Algorithm using Text Content of the Target Page", International Journal of Computer Applications, Vol. 85, No. 1, pp. 30-33, 2014.

[11]. Anuradha and A.K.Sharma, "A Novel Technique for Data Extraction from Hidden web Databases", International Journal of Computer Applications, Vol.15, No. 4, pp. 0975-8887, 2011.

[12]. Panagiotis G. Ipeirotis and Luis Gravano, "Distributed Search over the Hidden Web:Hierarchical Database Sampling and Selection", Proceedings of the 28th VLDB Conference, Hong Kong, China, pp. 394-405, 2002.

[13]. Lada A. Adamic and Bernardo A. Huberman, "The Web's Hidden Order", Proceedings of ACM , Vol. 44, No. 9, pp. 55-60, 2001.

[14]. Manuel Álvarez, Juan Raposo, Alberto Pan, Fidel Cacheda, Fernando Bellas and Víctor Carneiro, "Crawling the Content Hidden Behind Web Forms", International Conference of Computational Science and Its Application, Vol. 4706, pp. 322-333, 2007.