ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

Smartcrawler: A Two-stage Crawler Novel Approach for Web Crawling

Harsha Tiwary, Prof. Nita Dimble

Dept. of Computer Engineering, Flora Institute of Technology Pune, India

ABSTRACT: On the web, the non-indexed web pages are increasing rapidly. Many web crawlers have been developed to efficiently locate deep-web interfaces. But due to large no. of web resources and the dynamic nature of deep web achieving better result is a challenging issue. To solve this problem a two-stage framework called Smart-Crawler is proposed. This proposed system effectively searches the deep web. Smart-Crawler consists of two stages. The first stage in Smart-Crawler is reverse searching. It matches the users query with URL. The second stage is Incremental site prioritizing. It extracts the contents of the URLs and checks whether the query word is present in it or not. It also checks whether the extracted page is related to users profession or not. Then accordingly it classifies the pages as relevant and irrelevant. The relevant pages are then ranked using aho-corasick algorithm. The proposed crawler makes the search more personalized by searching the results according to users query and profession thereby improving the performance. User can also bookmark the links. The proposed crawler efficiently searches the deep-web interfaces. It produces the results that are better than the existing smart crawler.

KEYWORDS-Crawler, Deepweb, Feature selection, Sitefrequency, Two-stage crawler, Ranking, URL

I. INTRODUCTION

The World Wide Web (WWW) is a huge source of web pages. The World Wide Web is an information space where the web pages are identified by Uniform Resource Locators (URLs) which are interlinked by hypertext links and are accessible via the Internet. The WWW contains both static and dynamic web pages. But the no. of dynamic web pages in www are much higher than the static web pages. Each day there are many new web pages that keep on adding to the internet. The traditional search engine performs indexing on web pages. Since the internet contains a huge amount of data therefore, it is not possible to index each andevery web pages. Due to this, many relevant web pages remain unvisited by the traditionalsearch engine. Such web pages become the part of the deep web. To address this problem,smart-crawler has been proposed.

Smart-crawler is a crawler that efficiently searches the web for the relevant pages. Thesmart-crawler consists of two stages: Reverse searching and incremental-site prioritizing.In the first stage, smart crawler compares the URLs with the users query and classifies it asrelevant and irrelevant URL. In second stage, the contents of each relevant and irrelevantURL are extracted and the query word is searched in it.

The existing system was designed to perform normalized search. The user enters a query. The query is then processedand given to the seed collection. Here the seed collection uses the offline database. Theseed collection retrieves the data from the offline database and sends it to the adaptive sitelearner and site frontier. While extracting URLs it uses reverse searching. The adaptive sitelearner checks whether the site retrieved is duplicate or not. The site frontier is used forstoring the sites. The sites are then ranked and given to the site classifier. The site classifier
classifies the sites as relevant and irrelevant sites. For a given site, the incremental site stores
all the incoming and outgoing links into the link frontier. It then takes a link from the linkfrontier and extracts the contents for that link from the database. The process repeats until all the links are processed. The resultant links are

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

ranked. After ranking the resultant page is displayed to the end user. The resultant page contains the URL, title and description.

The existing system does not perform personalized search. Therefore, the chance of getting more relevant results in the result page is comparatively low. Also in the existingsystem, the two stages i.e. the reverse searching and incremental-site prioritizing does not work simultaneously. This affects the response time. However, these drawbacks have been resolved in the proposed system.

The proposed system has been designed to perform personalized search using smart-crawler. The main aim of this system is to provide more relevant results to the end user. Such type of search not only increases the tendency of getting more relevant data but also increases the performance.

II. MOTIVATION

The web pages are dynamic in nature. Each day there are many new web pages that keep on adding to the internet. The traditional search engine performs indexing on web page. Since the internet contains a huge amount of data therefore, it is not possible to index each and every web page. Such web pages remain unvisited by the traditional search engine. To address this problem, smart-crawler has been proposed. The proposed system performs personalized search using smart-crawler. Such type of search not only increases the tendency of getting more relevant data but also increases the performance.

III. REVIEW OF LITERATURE

Shukla [1] introduced a post query system. This system, filters out all irrelevant information which is not necessary according to the query entered by the user, and gives the expected results. The amount of data consumed by crawler while searching is huge. The crawler searches large amount of data that may contain lots of irrelevant information. Also a lot of time is wasted for searching relevant data from the huge amount of irrelevant results. This system takes more time to perform post-query.

Hatzi et al. [2] proposed a system which work on page refreshment policy. This policyminimizes the total Staleness of pages in the repository of a web crawler. In this system,threads are crawled concurrently. This retrieve pages from N web servers each time the user queries the system. Thus, it takes more time to get data from server.

Vijayarani et al. [3] introduced a system which is used for finding the useful informationfrom the large amount of data. Various data mining techniques are used to solve different types of search problems. It uses text mining for extracting information. The information is extracted from both structured data and unstructured data. After extracting information the system tries to find some patterns in it. Text mining techniques are used in various types of research domains like natural language processing, information retrieval, text classification and text clustering.

Gill et al. [4] introduced a system which is used in e-learning application. The e-Learninghas become popular. It has a learning paradigm which shifted the focus of entire world from instructor centric learning paradigm to learner centric approach. Its disadvantages are, this system worked only on E-learning application and does not give any domain classification.

Rahman [5] introduced a system which made use of topography order. A topography order is used to resolve the problem of over-information on the web or large domains. For this, the current information retrieval tools, especially search engines needed to be improved. Besides this much more intelligence needed to be incorporated into search tools. So that effective search, filtering processes and submission of relevant information can be performed.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

Cheng et al. [6] introduced that a task can be accomplished by performing only a smallnumber of queries, even in the worst case. It also explained that algorithms are asymptotically optimal i.e. it is impossible to improve their efficiency by more than a constant factor. The derivation of our upper and lower bound results reveals significant insight in the characteristics of the underlying problem. Extensive experiments confirmed that the proposed techniques worked very well on all the real datasets examined.

Shou et al. [7] has demonstrated the effectiveness of improving the quality of varioussearch services on the Internet. Proposed generalization aimed at striking a balance between two predictive metrics. These metrics finds the need of personalization and the privacy risk of exposing the generalized profile. This system uses two greedy algorithms, namely GreedyDP and GreedyIL. This system also provide an online prediction mechanism for deciding whether personalizing a query is beneficial. Kabisch et al.[8] proposed VisQI (VISual Query interface Integration system), a Deep Web integration system. VisQI is responsible for (1) Transforming web query interfaces into hierarchically structured representations. (2) Classifying query into application domains. (3) Matching the elements in different interfaces. Thus VisQI contains main solutions for the major challenges in building Deep Web integration systems.

Olston and Najork [9] introduced some steps for crawling the deep web. The steps are(1) Locating sources of web content. (2) Selection of releva1nt sources. (3) Extractingthe underlying content of deep web pages. Here the problem is, retrieval operation needs more time to crawl relevant results. Proposed system will perform reverse searching and incremental-site prioritizing to get relevant pages.

Chakrabarti et al. [10] proposed two hypertext mining programs. These programs allowed the crawler to evaluate the relevance of a hypertext document with respect to a specific topic. It presents an extensive focused-crawling experiment by using several topics at different levels of specificity. Focused crawling acquires relevant pages by focusing on a specific topic.

IV. SYSTEM OVERVIEW

The proposed architecture covers two types of search: search performed by the base paper i.e. normalized search and the proposed search i.e. the personalized search.

Starting with normalized search, first the user enters a query. This query is then processed in which the stop words that are present in the query are removed. This processed query is then send to the seed collection. The seed collection sends forwards this query to the online database i.e. the Google database. The Google database searches for all the relevant sites that contain the required query. The results obtained are then send back to the seed collection. The seed collection forwards these results to the reverse searching module. The reverse searching module compares the URLs of all the resultant sites with the query. If a URL contains the query word then it is classified as relevant URL else it is classified as irrelevant URL. Both these relevant and irrelevant URLs are send to the Incremental-site prioritizing module. This module extracts the contents of each URL and check whether the query word is present in its contents or not. If the query word is present in the content then it is classified as relevant document else it is classified as irrelevant document. All the relevant documents are then ranked according to the aho-corasick algorithm. After ranking the resultant page is then displayed to the end user. To find out which site is highly relevant for the specified query domain classification is also performed. The end user can also bookmark a page for its future reference.

In case of personalized search, user enters a query. The query is processed and sends to the seed collection. The seed collection forwards this processed query to the Google database. The Google database finds all the pages that are relevant to users query and users profession. The results obtained are send back to the seed collection. In this type of search, the reverse searching and incremental site prioritizing operates simultaneously. Therefore, the time required to refine the query is reduced. Here in reverse searching all the URLs are compared to the query and users profession. At the same time, Incremental site prioritizing extracts the contents of each URL and compares its contents with query and

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

users profession. All the resultant relevant pages are then ranked according to the aho-corasick algorithm. Domain classification can be performed on the resultant page. The end user can also bookmark a page for its future reference.

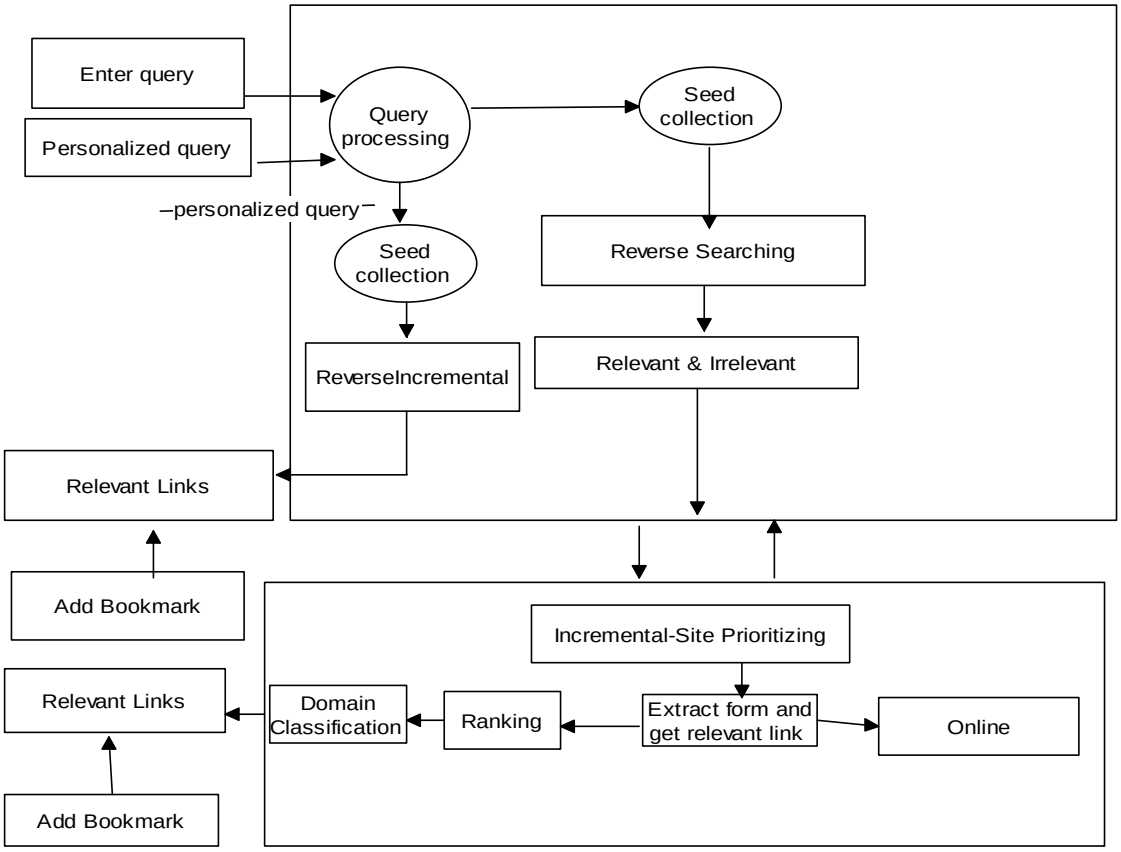

Fig. 01 System architecture

VI. MATHEMATICAL MODEL

Input : Query.
Output : Relevant links
Process

The feature space of deep web sites (*FSS*) is defined as:

FSS = U, A, T; ---------- **(1)**

where,
U, A, T are vectors corresponding to the feature context of URL, anchor, and text around URL of the deep web sites.

The feature space of links of a site with embedded forms (*FSL*) is defined as:

FSL = P, A, T ---------- **(2)**

where,
A and T are the same as defined in FSS
P is pattern which we are searching on extracted form.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

Each feature context can be represented as a vector of terms with a specific weight. The weight w of term t can be defined as:

$$w_{t,d} = tf_{t,d} \quad \text{----------} \quad (3)$$

where,
f (t,d) denotes the frequency of term t appears in document d.
and d can be U, P, A, or T.

VII.ALGORITHMS

Algorithm 1: Reverse Searching
Input: seed sites
Output: relevant sites and irrelevant site

Step 1: while candidate sites do
Step 2: pick a deep website
Step 3: links = extractLinks (link)
Step 4: page= compareUrl(links)
Step 5: relevant = classify (page)
Step 6: if relevant then
Step 7: listhq.add (page)
Step 8: return listhq
Step 9: else
Step 10: listLq.add(page)
Step 11: return listLq
Step 12: end

Algorithm 2: Incremental_site prioritizing
Input: list containing relevant and irrelevant sites
Output: High priority relevant Link

Step1: HQueue=list.CreateQueue (relevantLinks)
Step2: LQueue= list.CreateQueue(irrelevant Links)
Step3: while list is not empty do
Step4: if HQueue is empty then
Step5: HQueue.addAll (LQueue)
Step6: LQueue.clear ()
Step7: end
Step8: contents= content Extraction (link)
Step9: relevant=check(contents,query)
Step10: if(relevant) then
Step 11:add it into Hrelqueue.add(link)
Step 12:elseLrelqueue. add (link)
Step 13:end
Step 14:Apply the aho-corasick algorithm on the Hrelqueue
Step 15:end

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

VIII.EXPERIMENT RESULT

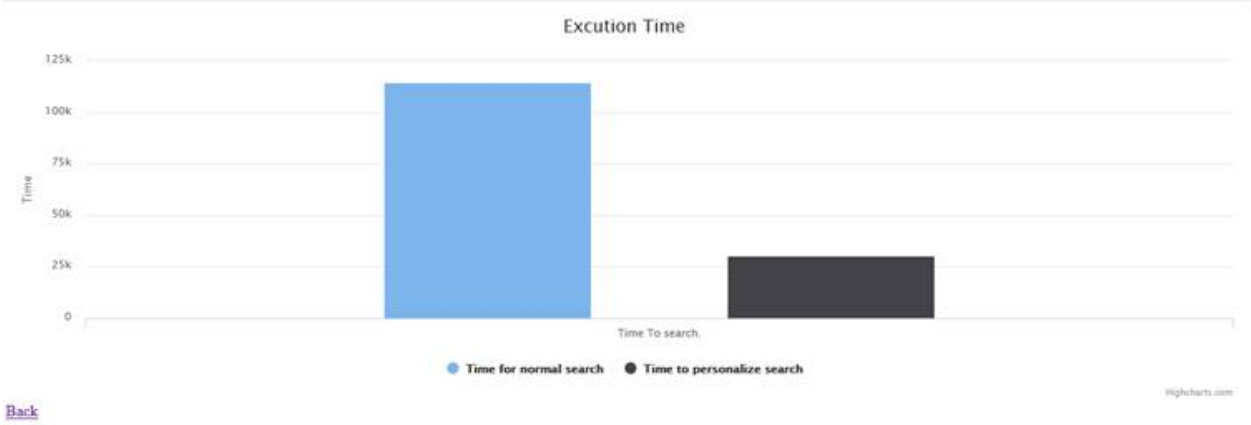

Fig.2] Time to search result

The figure shows the execution time taken by the normalized search and the personalized search.

2) Domain classification:

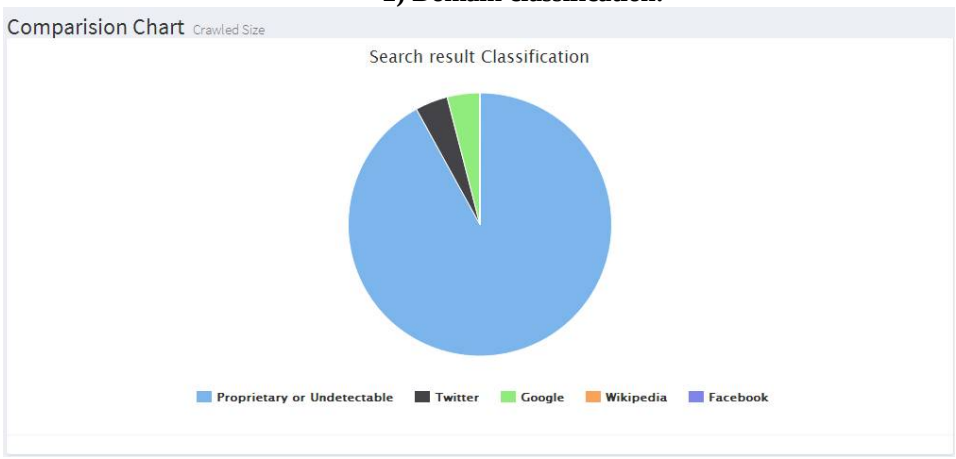

Fig.3] Graph shows domain classification.

This figure shows that no. Of links obtained after performing two-stage crawling. It will show how many links are obtained from which site for aentered query. The domain classification is given by the formula:
No. of links retrieved from a site=(no. of link of each domain / Total no. of relevantlink) *100

IX. CONCLUSION

The proposed system is the personalized version of the smart-crawler. It performs search according to users profession. The proposed smart crawler searches the deep web efficiently. In the proposed system, as the reverse search and incremental-site prioritizing are combined together and made to perform simultaneously therefore, there is an increase in its performance when compared to the normalized search of the smart-crawler. The pages are ranked according to the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 6, June 2018

aho-corasick algorithm. The domain classification is also performed. Log file is maintained where the users search history is saved. The user can also bookmark the pages for its future reference.

REFERENCES

[1] VishakhaShukla, Improving the Efficiency ofWeb Crawler by Integrating Pre Query Approach, Year- 2016
[2] Optimal Web Page Download Scheduling Policies for Green Web Crawling. Vassiliki Hatzi, B. BarlaCambazoglu, and IordanisKoutsopoulos. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 34, NO. 5, MAY 2016
[3] A Comparative Study of Hidden Web-Crawlers, International Journal of Computer Trends and Technology (IJCTT) Vol. 12,Sonali Gupta, Komal Kumar Bhatia Jun 2014.
[4] Personalization on E-Content Retrieval Based on Semantic Web Services A.B. Gil1, S. Rodrguez1, F. de la Prieta1 and De Paz J.F.1al.2013
[5] Search Engines going beyond Keyword Search: A Survey ,MahmudurRahman, 2013
[6] Optimal Algorithms for Crawling a Hidden Database in the Web Cheng Sheng Nan Zhang Yufei Tao Xin Jin. Proceedings of the VLDB Endowment, 5(11):11121123, 2012.
[7] Supporting Privacy Protection in Personalized Web Search, LidanShou, He Bai, Ke Chen, and Gang Chen,2012
[8] Deep web integration with visqi. Thomas Kabisch, Eduard C. Dragut, Clement Yu, and Ulf Leser. Proceedings of the VLDB Endowment, 3(1-2):16131616, 2010
[9] Web Crawling, Foundations and Trends in Information Retrieval, vol. 4, No. 3, pp.175246, 2010.Olston and M. Najork.
[10] Focused crawler: a new approach to topic-specific web resource discovery. Soumen Chakrabarti, Martin Van den Berg, and Byron Dom. 1999.