ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

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

Website: www.ijircce.com

Vol. 6, Issue 7, July 2018

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

Harsha Tiwary, Prof. Nita Dimble,

ME Student, Department of Computer Engineering, Flora Institute of Technology, Pune, India[1]

Department of Computer Engineering, Flora Institute of Technology, Pune, India[2]

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 developed 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 developed crawler efficiently searches the deep-web interfaces. It produces the results that are better than the existing smart crawler.

KEYWORDS: Crawler, URL, Site frequency, Deep Web, Two-stage crawler, Feature selection, Ranking.

I. INTRODUCTION

A web crawler also known as robot or spider is a massive download system for web pages. Web crawlers are used for a variety of purposes. Main components of web search engines, systems that assemble large web pages, point to them and allow users to publish queries in the index and find web pages that match queries. In the deep web there is growing interest in techniques that help you locate the deep interfaces efficiently. However, due to the large volume of web resources and the dynamic nature of the deep web pages, reaching a broad coverage and high efficiency is a challenge. We propose a two-stage framework, namely Smart Crawler, for efficient harvesting deep web interfaces. In the first stage, Smart Crawlerperforms Link based searching with the help of search engines, avoiding visiting a large number of pages. In second phase we are going to match form content, then classify the sites as relevant and irrelevant sites. Here we developed personalized search for efficient results and we are maintaining log for efficient time management.

II. MOTIVATION

It is challenging to get relevant pages for entered query. Web pages are usually sparsely distributed and keep on changingconstantly. To address this problem, previous work had proposed two types of crawlers, generic crawlers and focused crawlers. Besides efficiency, quality and coverage of relevant deep web sources are also challenging. To solve this we proposed two stage crawler that efficiently harvest deep web.

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.

DOI: 10.15680/IJIRCCE.2018.0607011

Hatzi et al. [2] proposed a system which works on page refreshment policy. This policy minimizes 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 information from 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-Learning has 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.

Cheng et al. [6] explained that a searching task can be accomplished by using small number 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 various search 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 FIT, Pune. Department of Computer Engineering 2018 3 exposing the generalized profile. This system uses two greedy algorithms, namely GreedyDP and GreedyIL. This system also provides 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) Extracting the 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. EXISTING SYSTEM

Existing systems were dealing with creation of a single profile per user, but conflict occurs when users interest varies for the same query. For example, a user is interested in banking exams so he enters a query “bank”. The result page contains the list of all the web pages which contain the query word such as blood bank money bank etc.. However, the user is interested in accounts of money bank and not in blood bank. In such case conflict occurs so we are dealing with negative preferences to obtain the fine grain between the interested results and not interested. Consider following two aspects:

1) Document-Based method:

These methods aim at capturing users clicking and browsing behaviors. It deals with the documents user has clicked on. Click through data in search engines can be thought of as triplets (q, r, c)

Where,

q = query
r = ranking
c = set of links clicked by user.

2) Concept-based methods:

These methods aim at capturing users conceptual needs. It deals with users browsed documents and search histories. User profiles are used to represent users interests and to infer their intentions for new queries.

DISADVANTAGES -

1) Deep-web interfaces.
2) Achieving wide coverage and high efficiency is a challenging issue.

V.SYSTEM OVERVIEW

In case of personalized search, user enters a query. The query is processed and send 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 users profession. Page ranking isperformed and high ranked results are then displayed on the result page. Domain classification is performed to show the users from which domain how many links are got. Here we are personalizing the search according to user profile so it is easy to get accurate result to user. User can bookmark the link for future use.

SYSTEM ARCHITECTURE

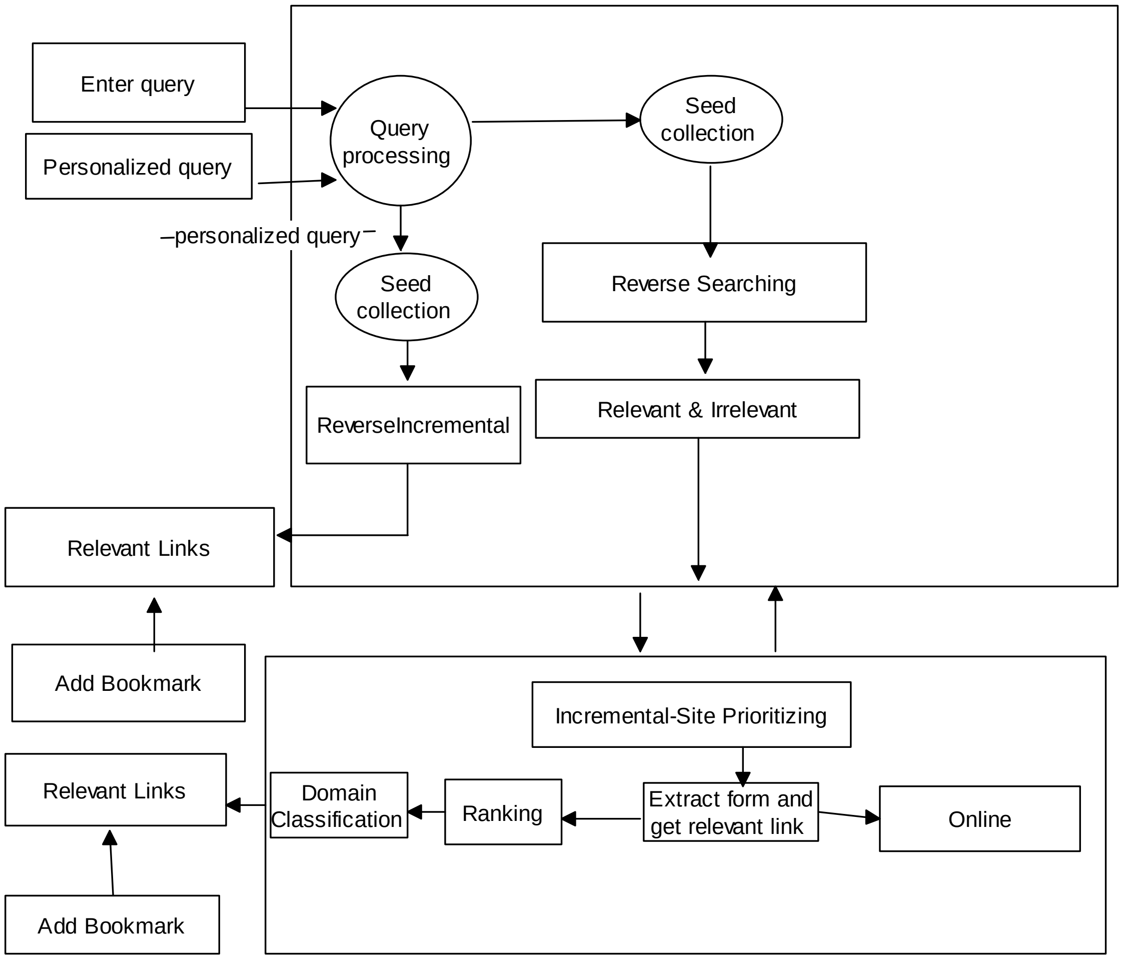

Fig. 01 System architecture

VI. MATHEMATICAL MODEL

Input given to the system is: - 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* and *P is pattern which we searching on extracted form.*
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:
$w_{t.d} = tf_{t,d}$ --------------------------------- **(3)**

Where $f_{t,d}$denotes the frequency of term *t* appears in document *d*.

VII. CONCLUSION

In this paper, the proposed crawler searches the deep-web pages efficiently. Due to the large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. Smart crawler gives better relevancy than other than crawler. *SmartCrawler* works in two phases: Reverse searching and Incremental-site prioritizing. The ranking helps to get relevant results. Log file is maintained to save time.Also the proposed system performs personalized search according to user profession.

REFERENCES

[1] Vassiliki Hatzi, B. Barla Cambazoglu and Iordanis Koutsopoulos, Optimal Web Page Download Scheduling Policies for Green Web Crawling. IEEE journal on selected areas in communications, vol. 34, no. 5, 2016.
[2] Vishakha Shukla, Improving the Efficiency of Web Crawler by Integrating Pre – Query Approach, 2016.
[3] Sonali Gupta and Komal Kumar Bhatia, A Comparative Study of Hidden Web-Crawlers, International Journal of Computer Trends and Technology (IJCTT) vol. 12, 2014.
[4] A.B. Gil1, S. Rodríguez1, F. de la Prieta1 and De Paz, Personalization on E-Content Retrieval Based on Semantic Web Services, 2013.
[5] Mahmudur Rahman, Search Engines going beyond Keyword Search: A Survey, 2013.
[6] Cheng Sheng, Nan Zhang, Yufei Tao and Xin Jin, Optimal Algorithms for Crawling a Hidden Database in the Web, Proceedings of the VLDB Endowment, 5(11):1112–1123, 2012.
[7] Lidan Shou, He Bai, Ke Chen and Gang Chen, Supporting Privacy Protection in Personalized Web Search, 2012.
[8] Thomas Kabisch, Eduard C. Dragut, Clement Yu and Ulf Leser, Deep web integration with visqi , Proceedings of the VLDB Endowment, 3(1-2):1613–1616, 2010.
[9] Olston and M. Najork, Web Crawling, Foundations and Trends in Information Retrieval, vol. 4, No. 3, pp. 175–246, 2010.
[10] Soumen Chakrabarti, Martin Van den Berg and Byron Dom, Focused crawler: a new approach to topic-specific web resource discovery, 1999.