Smart Web Crawler

[#1]Prof. Mrunal S. Bewoor, [#2]Puneet Arora, [#3]Abhishek Bist, [#4]Nishant Yadav

[1]msbewoor@bvucoep.edu.in

[2]puneet94arora@gmail.com

[3]Sanjubist17@gmail.com

[4]ynishant233@gmail.com

[#1234]Department of Computer Engineering,
BVUCOEP Pune,
Bharati Vidyapeeth Deemed University.

ABSTRACT

It's a challenging issue to achieve wide coverage and high efficiency Due to the large volume of web resources and the dynamic nature of deep web. So we propose A Smart Web Crawler which search and discovers Number of centre pages from deep web and focus its trajectory towards that topic in first stage i.e. Site locating due to which it avoids visiting a large number of pages. Smart Web Crawler ranks websites to prioritize highly relevant ones for a given topic. After searching centre pages in first stage it performs in-site exploration by excavating most relevant links with an adaptive link-ranking in second stage. Also there was confliction occurrences according to users interest due to single user so this drawback is also avoided in personalized Web Search engine. In this paper page refresh policy is used which re downloads the previously stored pages in the repository due to which HTTP requests are minimised so energy consumption and total staleness of pages are automatically decreases The existing crawler issues a large number of HTTP request to web server due to which there is more energy consumption and carbon footprint of web servers.

Keywords— **Deep web, two-stage crawler, carbon footprint, ranking, adaptive learning, personalization (profile based), staleness, greenness.**

I. INTRODUCTION

A crawler is a program that visits Web sites and reads their pages and other information in order to create entries for a search engine index. The major search engines on the Web all have such a program, which is also known as a "spider" or a "bot." Crawlers are typically programmed to visit sites that have been submitted by their owners as new or updated. Entire sites or specific pages can be selectively visited and indexed. Crawlers apparently gained the name because they crawl through a site a page at a time, following the links to other pages on the site until all pages have been read.

The crawler for the AltaVista search engine and its Web site is called Scooter. Scooter adheres to the rules of politeness for Web crawlers that are specified in the Standard for Robot Exclusion (SRE). It asks each server which files should be excluded from being indexed. It does not (or can not) go through firewalls. And it uses a special algorithm for waiting between successive server requests so that it doesn't affect response time for other users.

The deep (or hidden) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. Based on extrapolations from a study done at University of California, Berkeley, it is estimated that the deep web contains approximately 91,850 terabytes and the surface web is only about 167 terabytes in 2003 [1]. More recent studies estimated that 1.9 zettabytes were reached and 0.3 zettabytes were consumed worldwide in 2007 [2], [3]. An IDC report estimates that the total of all digital data created, replicated, and consumed will reach 6 zettabytes in 2014 [4]. A significant portion of this huge amount of data is estimated to be stored as structured or relational data in web databases — deep web makes up about 96% of all the content on the Internet, which is 500-550 times larger than the surface web [5], [6]. These data contain a vast amount of valuable information and entities such as Infomine [7], Clusty [8], BooksInPrint [9] may be interested in building an index of the deep web sources in a given domain (such as book). Because these entities cannot access the proprietary web indices of search engines (e.g., Google and Baidu), there is a need for an efficient

VI. OVERVIEW OF EXISTING SYSTEM

The existing system is a manual or semi automated system, i.e. The Textile Management System is the system that can directly sent to the shop and will purchase clothes whatever you wanted. The users are purchase dresses for festivals or by their need. They can spend time to purchase this by their choice like color, size, and designs, rate and so on. They But now in the world everyone is busy. They don't need time to spend for this. Because they can spend whole the day to purchase for their whole family. So we proposed the new system for web crawling.

.

II. PROPOSED SYSTEM

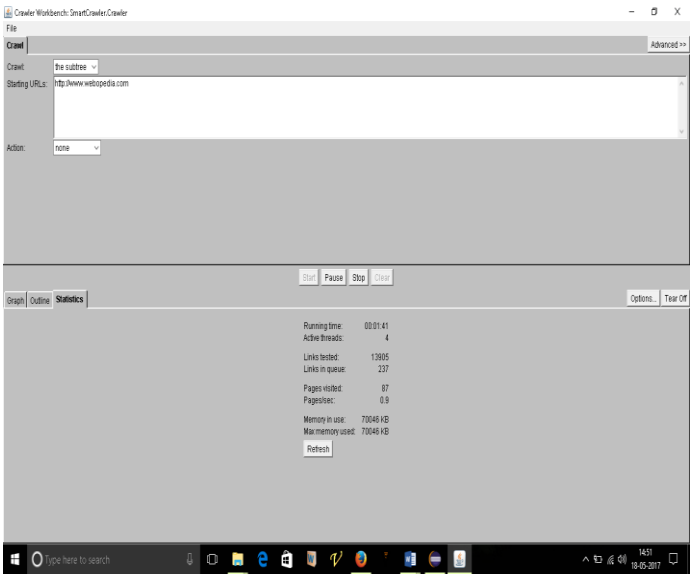
We propose a two-stage framework, namely Smart Crawler, for efficient harvesting deep web interfaces. In the first stage, Smart Crawler performs site-based searching for center pages with the help of search engines, avoiding visiting a large number of pages. To achieve more accurate results for a focused crawl, Smart Crawler ranks websites to prioritize highly relevant ones for a given topic. In the second stage, Smart Crawler achieves fast in-site searching by excavating most relevant links with an adaptive link-ranking. To eliminate bias on visiting some highly relevant links in hidden web directories, we design a link tree data structure to achieve wider coverage for a website. Our experimental results on a set of representative domains show the agility and accuracy of our proposed crawler framework, which efficiently retrieves deep-web interfaces from large-scale sites and achieves higher harvest rates than other crawlers. Propose an effective harvesting framework for deep-web interfaces, namely Smart-Crawler. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. Smart Crawler is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring. Smart Crawler performs site-based locating by reversely searching the known deep web sites for center pages, which can effectively find many data sources for sparse domains. By ranking collected sites and by focusing the crawling on a topic, Smart Crawler achieves more accurate results

Our project **web crawler with SEO analysis** actually providing information from that sites with chronological manner where student can find his/her information about results as well as college details and he/she also able to compare result with another student via proving unique number.With the help of **SEO** skill the most of search engine can able to find out data from our site and when any student wanted information about their result he/she need to type registration number or enrollment number, Roll number directly on the search engine etc..We are going to develop **Desktop application** for harvesting data and **Web application** for providing data online for student to search their data

III. RESULT ANALYSIS

Running Time	00:01:41
Active Threads	4
Links Tested	13905
Links In Queue	237
Pages Visited	87
Pages per Sec.	09
Memory In Use	70046 KB
Max. Memory Use	70046 KB

User enter the URL for search, smart crawler crawlers through source page of user ‘s given URL. It Extract all the sub URL’s level by level from source code via (DFS-BFS). It gives the result to user as in graphical and outlined form. Then it’s up to user to decide which URL to proceed with.



.

VII. Conclusion

.

Web crawler has met Author’s goals to make web scraping easy and automatic retraining possible. It has proved a convenient tool for extracting data from the web. Site Scraper has been tested over deferent domains with high effectiveness. Author believe that Site scraper can provide a robust and flexible solution for the problems of dealing with web data**.**

Author propose an effective deep web harvesting framework, namely Smart Crawler, for achieving both wide coverage and high efficiency for a focused crawler. Based on the observation that deep websites usually contain a few searchable forms and most of them are within a depth of three . crawler is divided into two stages: site locating and in-site exploring. The site locating stage helps achieve wide coverage of sites for a focused crawler, and the in-site exploring stage can efficiently perform searches for web forms within a site.

When user enter the URL for search, smart crawler crawlers through source page of user ‘s given URL. It Extract all the sub URL’s level by level from source code via (DFS-BFS). it gives the result to user as in graphical and outlined form. Then its up to user to decide which URL to proceed

IV. REFRENCES

[1] Peter Lyman and Hal R. Varian. How much information? 2003. Technical report, UC Berkeley, 2003.

[2] Roger E. Bohn and James E. Short. How much information? 2009 report on american consumers. Technical report, University of California, San Diego, 2009.

[3] Martin Hilbert. How much information is there in the ”information society”? Significance, 9(4):8–12, 2012.

[4] Idc worldwide predictions 2014: Battles for dominance – and survival – on the 3rd platform. http://www.idc.com/research/Predictions14/index.jsp, 2014.
[5] Michael K. Bergman. White paper: The deep web: Surfacing hidden value. Journal of electronic publishing, 7(1), 2001.
[6] Yeye He, Dong Xin, Venkatesh Ganti, Sriram Rajaraman, and Nirav Shah. Crawling deep web entity pages. In Proceedings of the sixth ACM international conferen