Web Crawler: A Crawler for Efficiently Retrieving Relevant Data

[1] Prabhu Alamkare [2] ShrikantGiri [3] Shubham Bardiya [4] PradeepGite
[1][2][3][4] JSPM'sRajarshi Shahu College of Engineering, Department of Computer Engineering
SavitribaiPhule Pune University, Pune, India.
[1] Prabhualamkare577@gmail.com [2] shrikantgiri622@gmail.com [3]bardiyashubham@gmail.com
[4] pradeep.gite10@ gmail.com

Abstract: **-The World Wide Web is a rapidly growing and changing information source. Due to the dynamic nature of the Web, it becomes harder to find relevant and recent information. WebCrawler are one of the most crucial part of the search engines to collect pages from the Web. WebCrawler is to download most relevant web pages from such a large web is still a major challenge in the field of Information Retrieval Systems. WebCrawler uses two-stage framework. In the first stage, WebCrawler performs site-based searching for visiting a large number of pages. In the second stage, WebCrawler achieves fast in-site searching by extracting most relevant links with an adaptive link-ranking. To achieve more accurate results WebCrawler ranks websites to prioritize highly relevant ones.**

***Keywords*: Deep web, two-stage crawler, feature selection, ranking, adaptive learning**

I. INTRODUCTION

A Web Crawler also known as a robot or a spider is a system for the bulk downloading of web pages. Web crawlers are used for a variety of purposes. Most prominently, they are one of the main components of web search engines, systems that assemble a corpus of web pages, index them, and allow users to issue queries against the index and find the web pages that match the queries.

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. Also known as a "spider" or a "bot" (short for "robot") .

Spider – programs like a browser to download the web page.
Crawler – programs automatically follow the links of web pages.
Robots - It had automated computer program can visit websites.

The deep(or hidden) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines.

Web Crawler is an indispensable part of search engine. A web crawler is a program that, given one or more seed URLs, downloads the web pages associated with these URLs, extracts any hyperlinks contained in them, and recursively continues to download the web pages identified by these hyperlinks. Web crawlers are an important component of web search engines, where they are used to collect the corpus of web pages indexed by the search engine. Moreover, they are used in many other applications that process large numbers of web pages, such as web data mining, comparison shopping engines, and so on.

Search engines are the primary gateways of information access on the Web. Today search engines are becoming necessity of most of the people in day to day life for navigation on internet or for finding anything. Search engine answer millions of queries every day.

1.1 Innovativeness of Purpose System:

An effective deep web harvesting framework, namely WebCrawler, for achieving both wide coverage and high efficiency.And main innovativeness of purpose system is it works in Two-Stage. Other innovativeness is it can work efficiently and accurately.

A two-stage framework to address the problem of searching for hidden-web resources. To efficiently and effectively discover deep web data sources, WebCrawler is designed with two stage architecture, site locating and in-site exploring. Innovativeness is adaptive learning algorithm that performs online feature selection and uses these features to automatically construct link rankers and site locating technique employs a reverse searching technique and site prioritizing technique for relevant sites, achieving more data sources.

II. EVALUATION OF SYSTEM

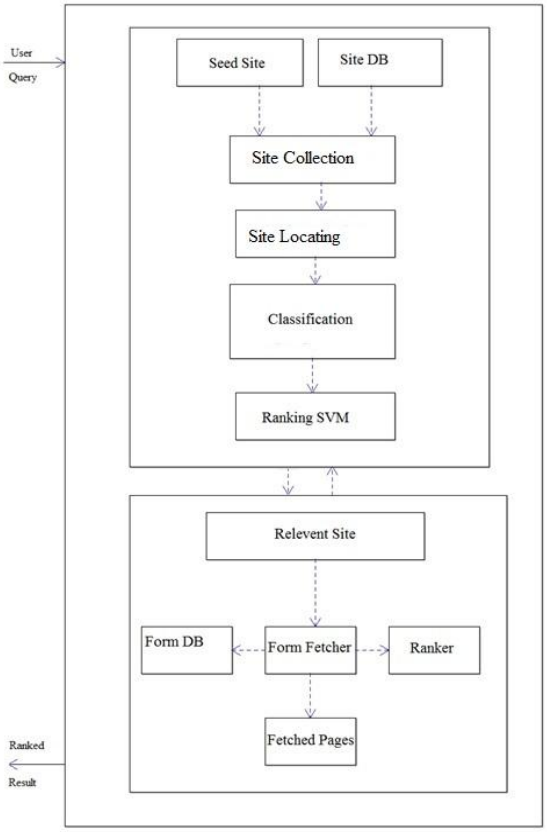

Fig 1: System Architecture

Site Locating: The site locating stage find relevant sites for a given topic, consisting of site collecting, site ranking, and site classification.

In-Site Exploring: Once a site is regarded as topic relevant, in-site exploring is performed to find searchable forms. Links within a site are prioritized with Link Ranker and Form Classifier classifies searchable forms.

Seed Site: site locating stage starts with a seed set of sites in a site database. Seeds sites are candidate sites given for *WebCrawler* to start crawling, which begins by following URLs from chosen seed sites to explore other pages and other domains

Adaptive Learning**:** WebCrawler has an adaptive learning strategy that updates and leverages information collected successfully during crawling.

Site Ranking**:** WebCrawlerranks site URLs to prioritize potential deep sites of a given topic. To this end, two features, site similarity and site frequency, are considered for ranking.

Fetched pages**:** Links of a site are stored in Link Frontier and corresponding pages are fetched and embedded forms are classified by Form Classifier to find searchable forms.

Incremental site prioritizing to make crawling process that achieve broad coverage on websites. Feature that WebCrawler should provide is distributed, scalable, performance and efficiency, quality, freshness, and extensible.

While crawling, *SmartCrawler*follows the out-ofsitelinks of relevant sites. To accurately classify out-of-site links, Site Frontier utilizes two queues to save unvisited sites. The high priority queue is for out-of-site links that are classified as relevant by Site Classifier and are judged by Form Classifier to contain searchable forms. The low priority queue is for out-ofsitelinks that only judged as relevant by Site Classifier. For each level, Site Ranker assigns relevant scores for prioritizing sites.

III. ALGORITHM

Algorithm 1: Reverse searching for more sites.

Input**:** seed sites and harvested deep websites
Output::relevant sites

```
while # of candidate sites less than a threshold do
// pick a deep website
site= getDeepWebSite(siteDatabase, seedSites)
resultPage= reverseSearch(site)
links= extractLinks(resultPage)
    foreach link in links do
    page = downloadPage(link)
    relevant = classify(page)
            if relevant then
            relevantSites= extractUnvisitedSite(page)
             Output relevantSites
             end
     end
end
```

Algorithm 2: Incremental Site Prioritizing.
Input : SiteFrontier
Output: Searchable forms and out-of-site links

```
HQueue=SiteFrontier.CreateQueue(HighPriority)
LQueue=SiteFrontier.CreateQueue
```

```
(LowPriority)
while siteFrontier is not empty do
    if HQueue is empty then
        HQueue.addAll(LQueue)
        LQueue.clear()
    end
    Site= HQueue.poll()
    Relevant= classifySite(site)
    if relevant then
        performInSiteExploring(site)
        Output forms and OutOfSiteLinks
        siteRanker.rank(OutOfSiteLinks)
        if forms is not empty then
            HQueue.add(OutOfSiteLinks)
        end
    else
        LQueue.add(OutOfSiteLinks)
    end
end
```

IV. TECHNICAL DETAILS

WebCrawler has an adaptive learning strategy that updates and leverages information collected successfully during crawling. As shown in figure both Site Ranker and Link Ranker are controlled by adaptive learners. Site Ranker and Link Ranker are updated. Finally, Site Ranker re-ranks sites in Site Frontier and Link Ranker updates the relevance of links in Link Frontier.

Figure illustrates the adaptive learning process that is invoked periodically. For instance, the crawler has visited a pre-defined number of deep web sites or fetched a pre-defined number of forms.

When a site crawling is completed, feature of the site is selected for updating *FSS* if the site contains relevant forms. During in-site exploring, features of links containing new forms are extracted for updating *FSL*.

4.1 Site Ranking

WebCrawler ranks site URLs to prioritize potential deep sites of a given topic. To this end, two features, site similarity and site frequency, are considered for ranking. Site similarity measures the topic similarity between a new site and known deep web sites. Sitefrequency is the frequency of a site to appear in othersites, which indicates the popularity and authority ofthe site a high frequency site is potentially more important.

4.2Link Ranking

For prioritizing links of a site, the link similarity is computed similarly to the site similarity described above. The difference includes: 1) link prioritizing is based on the feature space of links with searchable forms,

2) For URL feature *U*, only path part is considered since all links have the same domain and 3) the frequency of links is not considered in link ranking.

V. CONCLUSION

An effective harvesting framework for deep-web interfaces, namely WebCrawler. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains efficient crawling. Crawler consisting of two stages: efficient site locating and balanced in-site exploring 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 achieves more accurate results. The in-site exploring stage uses adaptive link-ranking to search within a site.

VI. ACKNOWLEDGEMENT

We would like to acknowledge the guidance of Mr. Santosh B. Javheri for his support and inspiration throughout the various stages of this paper. We sincerely appreciate the help and advice given by him which went a long way in helping us understanding the key concept of this paper.

REFERENCES:

[1]. WenwenLia, ChaoweiYanga and ChongjunYangb,"An active crawler for discovering geospatial Web services and their distribution pattern –Vol. 24, No. 8, August 2010.

[2]. MahmudurRahman,"Search Engines going beyond Keyword Search",School of Computing and Information Sciences Florida International University, Miami, FL 33199,Volume 75 - No. 17, August 2013.

[3]. Trupti V. Udapure, Ravindra D. Kale, Rajesh C. Dharmik,"Study of Web Crawler and its Different Types" ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014)

[4]. A.B. Gil, S. Rodríguez, F. de la Prieta and De Paz J.F,"Personalization on E-Content Retrieval Based on Semantic Web Services.

[5]. Pavalam S M, S V Kashmir Raja, Felix K Akorli3 and Jawahar M,"A Survey of Web Crawler Algorithms" National University of Rwanda Huye, RWANDA,Vol. 8, Issue 6, No 1, November 2011.

[6]. Ms. Pallavi Wadibhasme1, Prof. NitinShivale ,"Survey on – Self Adaptive Focused Crawler,Issue 6, No 1, November 2013.

[7]. Paolo Boldi_ Bruno Codenotti† Massimo Santini‡ SebastianoVigna“ A Scalable Fully Distributed Web Crawler”

[8]. Tiffany Ya TANG and Gordon MCCALL Smart Recommendation for an Evolving E-Learning System

[9] .Feng Zhao, Jingyu Zhou, Chang Nie, Heqing SmartCrawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces. IEEE Transactions on Services Computing Volume: PP Year: 2015

[10]. RajashreeShettar, Dr.Shobha G. “Web Crawler On Client Machine” Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol II IMECS 2008, 19-21 March, 2008.