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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

An Advanced Integrated Crawling Architecture for Extracting Topic Specific and Hidden Deep Web Entries

Vrutuja P.Pande, Prof. Pratap Singh

PG Student, Dept. of Computer Engineering, Institute of Knowledge College of Engineering, Pune University, India

Assistant Professor, Dept. of Computer Engineering, Institute of Knowledge College of Engineering, Pune University, India

ABSTRACT: Deep Web is a vague description of the internet not necessarily accessible to search engines. The Deep Web is a part of the internet not accessible to link-crawling search engines like Google. The only way a user can access this portion of the internet is by typing a directed query into a web search form, thereby retrieving content within a database that is not linked. In layman's terms, the only way to access the Deep Web is by conducting a search that is within a particular website. The Surface Web is the internet that can be found via link-crawling techniques; link-crawling means linked data can be found via a hyperlink from the homepage of a domain. Google can find this Surface Web data. Surface Web search engines (Google/Bing/Yahoo!) can lead you to websites that have unstructured Deep Web content. To reduce the time and cost of crawling and Performing an exhaustive crawl is a challenging question. Additionally, capturing the model of a modern web application and extracting data from it automatically is another open question. We propose a intelligent crawler for deep web harvesting and extracting topic specific Hidden web entries which can be referred as concept based semantic search engine. The crawler not only aims to crawl the World Wide Web and bring back data but also aims to perform an initial data analysis of unnecessary data before it Stores the data. The proposed architectures extract the deep web data and improve the efficiency of the Concept Based Semantic Search Engine and achieve wide coverage and high efficiency.

KEYWORDS: Deep web, crawling. Hidden web entries Concept Based Semantic Search Engine

I.INTRODUCTION

Main component of search engine is Web Crawler. Web crawler is automatic program that browses the web & download information. Search engine uses this downloaded web pages to store in repository, this repository is used to generate the results of search. Web crawler is also used in many other services like search engines, digital library online marketing, web data mining & search for personal information such as emails, numbers, address for marketing and spam mails. Size of web is increasing at very high rate. Google announced that Google have revealed one trillion unique URL in May 2009.Over 109.5 million Websites operating.

Yuan and Harms in 2002 review the log file of web server at department of computer Science at University of Alberta and find that maximum 40.6% total web traffic is due to hits by web crawler. Crawlers are not actual user so the heavy crawling traffic is not good for websites and network. It is most horrible for websites new in business. Bal and Nath in 2010 perform experiment on web. They download home pages of 100 different web sites daily for 30 days.

They find 52% pages change every day. 48% of web pages don't change daily.

Web managers can direct web crawler by using Robots Exclusion Protocol. It is a convention to avoid Web Crawler from accessing all or part of a Website which is openly viewable. Protocol use "robot.txt" file. Web crawler fined this file at server and follows

DOI: 10.15680/IJIRCCE.2016. 0404294

the directions in robot.txt. Robot.txt file is maintained by Web manager.
A basic layout of what this looks like is shown below

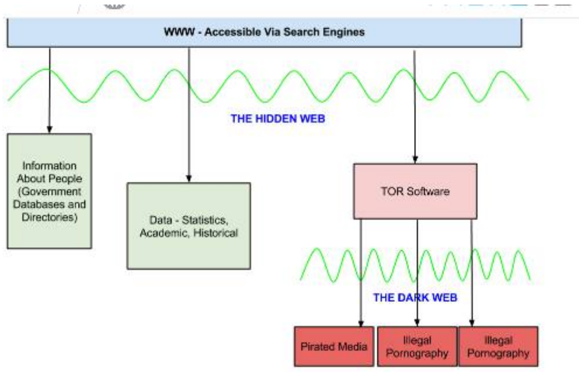

WWW - Accessible Via Search Engines
THE HIDDEN WEB
Information About People (Government Databases and Directories)
Data - Statistics, Academic, Historical
TOR Software
THE DARK WEB
Pirated Media
Illegal Pornography
Illegal Pornography

Fig 1 Example of deep web data

Methods of accessing these different parts of the deep web are determined by the data that you want to get at. The tools used to navigate the deep web are outlined here.

- Databases – Information about people, census data, climate data, world information and other searchable information that could be stored in a table format.
- Journals and Books – Information contained in a digital format that is either stored in a format not accessible by web crawlers or exists behind a paid gateway. These files need to be downloaded and opened on a PC.
- Tor Network – Sites that want to remain hidden, and typically include things like illegal porn, stolen personal data, drug contacts, anonymous political dissidents, terrorists, and more.

A. Problem Statement

The hidden Web provides access to huge and rapidly growing data repositories on the Web. 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

II. RELATED LITERATURE SURVEY

Most of the researchers focus on the architecture of the algorithms that are able to collect the most relevant pages with the corresponding topic of interest. The term focused crawling was originally introduced by (Chakrabarti, Berg, & Dom, 1999) which indicates the crawl of topic-specific web pages. In order to save hardware and network resources, a focused web crawler analyzes the crawled pages to find links that are likely to be most relevant for the crawl and ignore the irrelevant clusters of the web. Chakrabarti, Berg and Dom (1999) descriped a focused web crawler with three components, a classifier to evaluate the web page relevance to the chosen topic, a distiller to identify the relevant nodes using few link layers, and a reconfigurable crawler that is governed by the classifier and distiller. Web page credtis problem was addressed by (Diligenti, Coetzee, Lawrence, Giles and Gori, 2000), in which the crawl paths chosen based on the number of pages and their values. They use context graph to capture the link hierarchies within which valuable pages occur and provide reverse crawling capabilities for more exhaustive search. Suel and Shkapenyuk (2002) described the architecture and implementation of optimized distributed web crawler which runs on multiple work stations. CROSSMARC approach was introduced by (Karkaletsis, Stamatakis, Horlock, Grover and Curran, 2003). CROSSMARC employs language techniques and machine learning for multi-lingual information extraction and consists of three main components: site navigator to traverse web pages and forward the collected information to (Page filtering) and (Link scoring). Baeza-Yates (2005) highlighted that crawlers in the search engine are responsible for generating the structured data and they are able to optimize the retrieving process using focused web crawler for better search results. Castillo (2005) Designed a new model for web crawler, which was integrated with the search engine project (WIRE) and provided an access to metadata that enables the web crawling process.

III. EXISTING SYSTEM

A Web crawler is an Internet bot which systematically browses the World Wide Web, typically for the purpose of indexing. Web and some other sites use Web crawling or speeding software to update their web content or indexes of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine which indexes the downloaded pages so the users can search much more efficiently. Crawlers consume resources on the systems they visit and often visit sites without tacit approval. Issues of schedule, load, and "politeness" come into play when large collections of pages are accessed. Mechanisms exist for public sites not wishing to be crawled to make this known to the crawling agent. As the number of pages on the internet is extremely large, even the largest crawler fall short of making a complete index Crawlers can validate hyperlinks and HTML code. They can also be used for web scraping (see also data-driven programming).

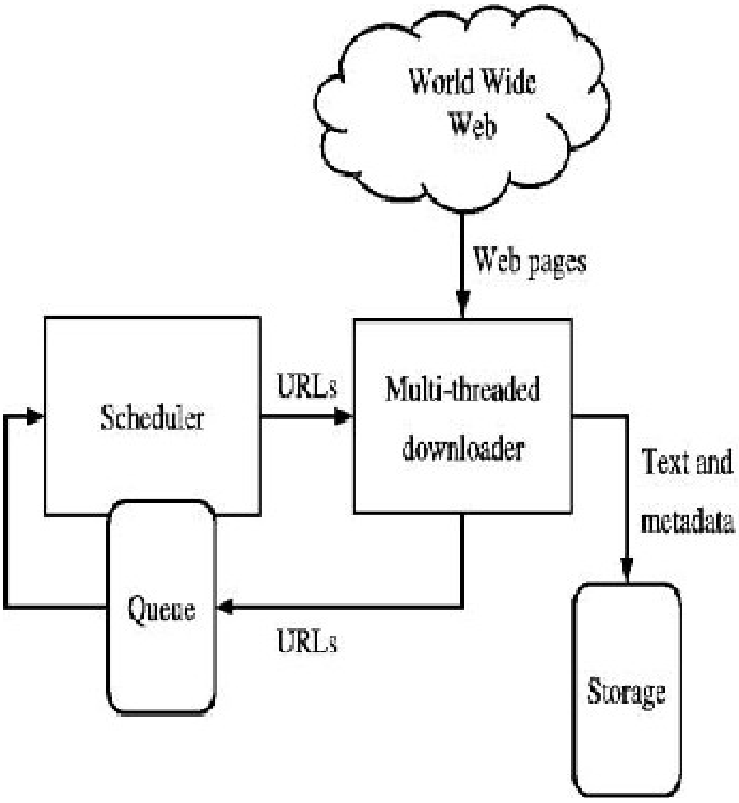


Fig 2.Existing web crawler architecture

IV. PROPOSED SYSTEM

A. System Architecture

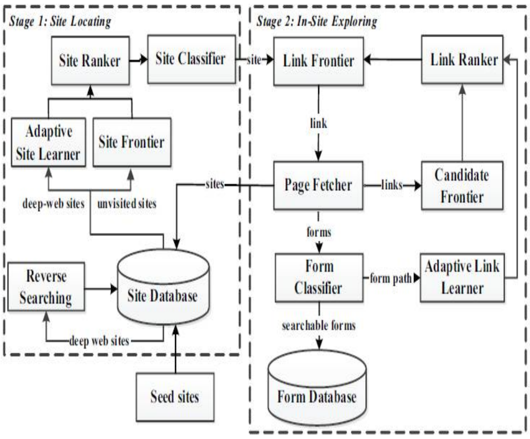


Fig. 3.1 .Architecture for extracting deep web

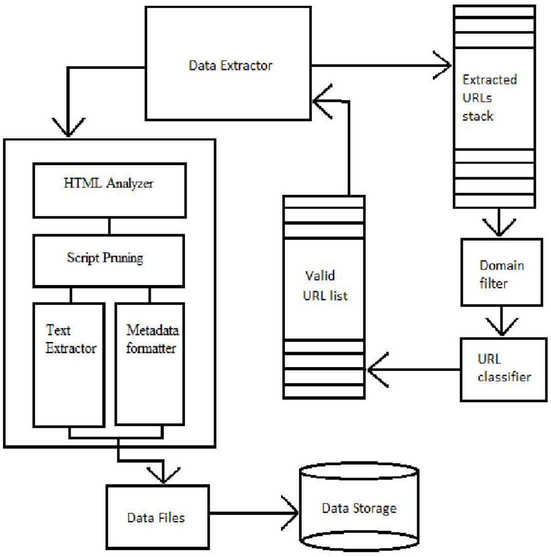


Fig 3.2 Architecture for topic specific hidden web entries

B. Proposed Algorithm

Algorithm 1: extracting domain specific entries

```
For(each url in urllist)
{
    If (MIME type == html or txt)
    {
        If(url == given domain)
        {
            Extract data();
```

```
      Extract urls();
      Cleaner.script()
      Extract text();
      Extract metadata();
      Save to disk();
    }
  }
}
```

Algorithm 2: 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)
    resultP age = reverseSearch(site)
    links = extractLinks(resultP age)
    foreachlink in links do
        page = downloadPage(link)
        relevant = classify(page)
        if relevant then
            relevantSites= extractUnvisitedSite(page)
            Output relevantSites
        end
    end
end
```

Algorithm 3: 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)
```

19 end
20 end
21 end

C. Data Flow diagram

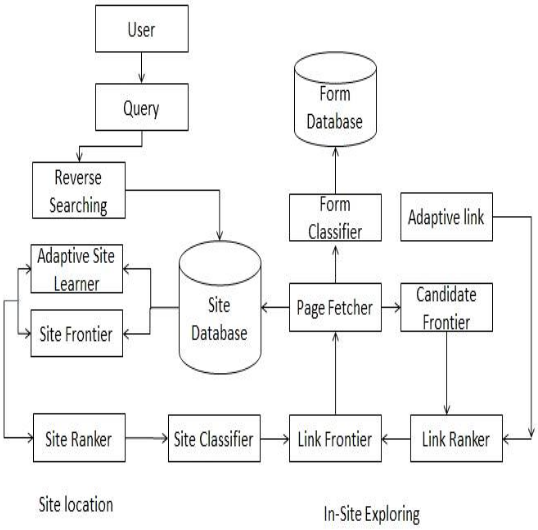


Figure 4 System Data Flow diagram

V. MATHEMATICAL MODEL

The mathematical model is a description of a system using mathematical concepts and languages. For system explanation and study the different components effects, set theory is used.
A. Set Theory:
Let S is the Whole System Consist of
S= {Q, D, F}.
Where Q is set of query entered by user.
Q={q1, q2, q3,…..qn}.
D = Data set.
F = Functions used.
F={RS, ASL, SF, SR, SC}
RS = Reverse searching.
ASL = Adaptive site learner
SF = Site Frontier
SR = Site Ranker
SC = Site Classifier

VI .IMPLEMENTATION DETAILS

In this section we present the input, expected result and environment used for implementation

A. Input

For this implementation, we use the input as text file, or data to be retrieved, which is further spitted into number of blocks.

B. Hardware and Software Used

Hardware Requirements:

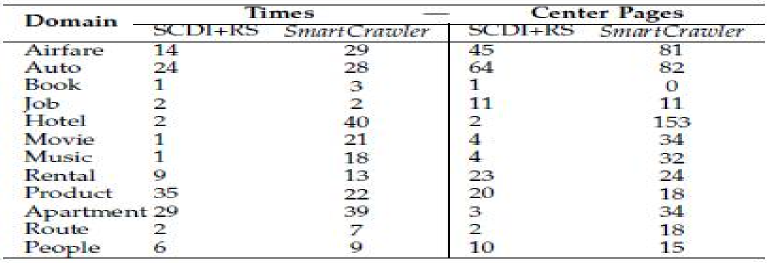
- Processor : Pentium IV
- Speed :1.1 GHZ
- RAM :25MB(min)
- Hard Disk :20 GB
- Key Board :Standard Windows Keyboard
- Mouse :Two or Three Button Mouse
- Monitor : SVGA

Software Requirements:

- Operating system : Windows, XP/7.
- Coding Language : JAVA
- IDE :Net beans 7.4
- Database :MYSQL

C. Experimental Results

We have implemented *praposed crawler* in Java and evaluated our approach over 12 different domains described in Table below



Domain	Times		Center Pages	
	SCDI+RS	*SmartCrawler*	SCDI+RS	*SmartCrawler*
Airfare	14	29	45	81
Auto	24	28	64	82
Book	1	3	1	0
Job	2	2	11	11
Hotel	2	40	2	153
Movie	1	21	4	34
Music	1	18	4	32
Rental	9	13	23	24
Product	35	22	20	18
Apartment	29	39	3	34
Route	2	7	2	18
People	6	9	10	15

Table1. The comparison of the number of triggered times and center pages found for reverse searching

Smart- Crawler leverages learning results for site ranking and link ranking. During in-site searching, more stop criteria are specified to avoid unproductive crawling in proposed *Crawler*. Thus our proposed architecture explores more central pages than site based crawler for deep web interfaces and time required for crawling is also significantly low.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

8

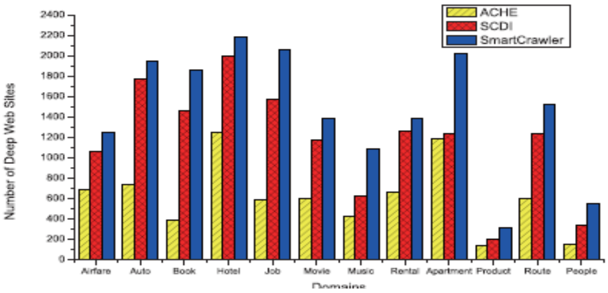


Fig5 The numbers of relevant deep websites harvested by ACHE, SCDI and *proposed smart Crawler*

Fig 5 shows that *proposed Crawler* finds more relevant deep websites than ACHE and SCDI for all Domains. Figure illustrates that *proposed Crawler* consistently harvests more relevant forms than both ACHE and SCDI. SCDI is significantly better than ACHE because our two-stage framework can quickly discover relevant sites rather than being trapped by irrelevant sites. By prioritizing sites and in-site links, *Crawler* harvests more deep websites than SCDI, because potential deep websites are visited earlier and unproductive links in in-site searching are avoided. Most of bars present a similar trend in Figure 5 because the harvested sites are often proportional to harvested searchable forms.

VII. CONCLUSION

We conclude that our approach achieve wide coverage for deep web interfaces and maintains highly efficient crawling. Our crawler achieves more accurate results. We built a intelligent crawler to serve the needs of the Concept Based Semantic Search Engine.. Our experimental results on representative set of domains show the effectiveness of the proposed two-stage crawler, which achieves higher harvest rates than other crawlers. Our system selects work very carefully from the crawl frontier. A consequence of the resulting efficiency is that it is feasible to crawl to a greater depth than would otherwise be possible. This may result in the discovery of some high-quality information resources that might have otherwise been overlooked. With the filtered text files generated bythe intelligent Crawler the Semantic Search Engine was able to identify concepts from the data quickly and in a much more efficient way. Thus we were able to improve the efficiency of the Concept Based Semantic Search Engine.

VIII. FUTURE SCOPE

The future scope of our project is that we are planning to design Post-query technique that identifies searchable forms by submitting the probing queries to forms and analyzing the result pages.

REFERENCES

[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 conference on Web search and data mining*, pages 355–364. ACM, 2013.
[7] Infomine. UC Riverside library. http://lib-www.ucr.edu/, 2014.
[8] Clusty’s searchable database dirctory. http://www.clusty.com/, 2009.

[9] Booksinprint. Books in print and global books in print access. http://booksinprint.com/, 2015.

[10] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the web. In *CIDR*, pages 44–55, 2005.

[11] Denis Shestakov. Databases on the web: national web domain survey. In *Proceedings of the 15th Symposium on International Database Engineering & Applications*, pages 179–184. ACM, 2011.

[12] Denis Shestakov and Tapio Salakoski. Host-ip clustering technique for deep web characterization. In *Proceedings of the 12th International Asia-Pacific Web Conference (APWEB)*, pages
378–380. IEEE, 2010.

[13] Denis Shestakov and Tapio Salakoski. On estimating the scale of national deep web. In *Database and Expert Systems Applications*, pages 780–789. Springer, 2007.