ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 6, June 2016

# Intelligent Crawler for Deep Web Harvesting using Page Rank Mechanism

**Ms. Sneha Avinash Ghumatkar[a]*, Dr. Archana Lomte[b],Dr. Gayatri Bhandari**

[ab]Computer Engineering Department JSPM'S Bhivrabai Sawant Institute of Technology & Research Wagholi, Pune, Savitribai Phule Pune University,India

***Abstract*— Site pages accessible in the web are developing massively now days. In such a circumstance looking more pertinent data in the Internet is a hard undertaking. Big data is holed up behind question frames, this data interface to undetermined databases containing top notch organized information. Customary web crawlers can't get to and record this concealed part of the Web. Retraining this concealed data from web is extremely testing assignment. In this way, we show two sorts of structure, to be particular Smart-Crawler, for effectively gathering significant web interfaces. In the principle organize that is webpage discovering, center pages are looked with the help of web records which consequently swear off passing by incalculable. To achieve more unbendable results for a connected with killjoy, Smart-Crawler positions locales to compose incredibly suited ones for a given point. In the second stage, flexible association - exhuming in order to position fulfills speedy in - site looking for most suited associations. To slaughter inclination on passing by some exceptionally related associations in covered web lists, we arrange an association tree data structure to perform gigantic degree for a website. The Smart-Crawler strategies simply consider a url. So we use Smart Search framework for request using page rank estimation. The trial results on a course of action of representative spaces exhibit the inclination and precision of proposed crawler framework, which capably recoups significant web interfaces from broad - scale destinations and access higher harvest rates than various crawlers.**

***Index Terms*— Clustering, Classification and Association Rules, Data Mining.**

## I. INTRODUCTION

Fundamentally, Crawler means, it slithers around the ground. In web creeping, the crawler slithers around the web - pages, gathers and classifies data on the World Wide Web. The crawler contains of three sections: First is the insect, additionally called as crawler. The pages are gone by creepy crawly, get the data and after that take after the connections in different pages inside of a site. The wok comes back to crept site over normal interim of time. The data found in the primary stage will be dependent on the second stage, the record. It is likewise well - known as index. The list is similar to a database, containing every duplicate of web - page that crawler finds. In the event that a web - page changes then the duplicate is overhauled in the database with new data. Programming is third part. Level the website pages in requested of most significant once this project shift a great many site pages enrolled in the list to discover matches to pursuit them.

Site pages enlisted in the list to discover matches to pursuit and level them all together of what it accepts as generally applicable.

Profound web likewise called as dim web or imperceptible web. Profound web are the substance on the web which is not ordered in a web crawler. It is various sites that are openly accessible yet shroud the IP locations of a server that keep running on them. Therefore client can be gone to by them, however it is hard to discover who are behind those destinations. Profound web is something you can't situate with a solitary pursuit.

To find profound web interfaces is troublesome errand, as they are not recorded by any web search tools. They are generally keep continually changing and once in a while circulated. To manage above issue, past work has proposed two sorts of crawlers which are engaged crawlers and non specific crawlers. Bland crawler brings all the searchable structures and don't focus on a particular subject though Focused crawlers are the crawler which concentrates on a particular theme. Versatile crawler for concealed web passages (ACHE) and Form - centered crawler (FFC) plans to productively and naturally distinguish different structures in the same space [1]. The FFC fundamental segments are connection, page, structure classifiers and wilderness chief for centered slithering of web - shapes. Throb broadens the engaged technique of FFC with extra parts a versatile connection learner and structure sifting. The connection classifiers assume a focal part to achieve higher creeping proficiency than the best - first crawler. The precision of centered crawlers is low as far as recovering significant structures. Case in point, an examination directed for database spaces, it has been demonstrated that the curacy of Form - Focused Crawler is around 16 percent. Hence it is

***Manuscript received June, 2016.***
***Ms. Sneha Avinash Ghumatkar**, Computer Engineering Department JSPM'S Bhivrabai Sawant Institute of Technology & Research Wagholi, Pune, Savitribai Phule Pune University, India, Pune, India,*
***Dr. Archana Lomte**, Computer Engineering Department JSPM'S Bhivrabai Sawant Institute of Technology & Research Wagholi, Pune, Savitribai Phule Pune University, India.*

important to create brilliant crawler that can rapidly find significant substance from the profound web however much as could reasonably be expected [2].

Two systems for proficiently gathering profound web named Smart-Crawler are outlined in this undertaking. Both strategies perform a propelled level of information investigation and information separated from the web. These systems are separated into phases of two: in-site investigating and Site finding. In the phase of in the first place, these methods perform with the assistance of web search tools for the webpage - based hunting down focus pages, abstaining from going by countless. To accomplish more definite results for a focused on creep, Ranks sites for Smart-Crawler to set up very pertinent once for a given theme. In the phase of second, Smart-Crawler accomplishes quick in - site looking to unearth most important connections with a versatile connection - positioning.

We propose a Smart-Crawler method for url based collecting profound web interfaces. SmartSearch strategy for inquiries based reaping profound web interfaces utilizing page rank calculation.

**Existing System**

To discover Large measure of data that is digged behind profound web interfaces is a test and part of work are proposed to do as such.

The principal Web crawler presented by Matthew dim implemented the globe Wide web Wanderer. The Wanderer was composed in Perl and kept running on one machine. It had been utilized till 1996 to assemble insights concerning the development of the on the web. In addition, the pages crept by the Wanderer were put into partner list (the ―Wandex), along these lines offering ascend to the main PC software engineer on the on the web, Gregorian extra crawler-based web Search motors got to be accessible In year 1993, schedule month 3: Jump Station (executed by Jonathan Fletcher; the arranging has not been composed up), Also the World Wide Web Worm [90], and RBSE creepy crawly. WebCrawler joined the field in Apr 1994, and MOM insect was outlined a comparable year. This original of crawler's distinguished some of the characterizing issues in web crawler style.

**Existing Advantages**

- It is simple architectural approach.

**Existing Disadvantages**

- It is just focused on homepage URL's and not considers deep URL's because of their dynamic nature.

**Proposed System**

This paper proposes a new crawler that provides user friendly, efficient, fast, well structured search results. Smart-Crawler, for productive collecting profound web interfaces. We propose a two-stage structure, It contains two stages. 1) Smart-Crawler and 2) Smart Search. In the principal stage, Smart-Crawler performs with help of web indexes to website based hunting down focus pages to abstaining from going by an expansive number of pages. To accomplish more precise results for an engaged slither, Smart-Crawler positions sites to exceedingly organize applicable ones for a given subject. In the second stage, Smart-Crawler accomplishes quick in-site seeking by revealing most important connections with a versatile connection positioning. Shrewd Search procedure utilized for rank sites as a part of clients inquiry question results utilizing Page Rank calculation.

**Proposed System Advantages**

- Our proposed work focused URL with Queries (Keywords).

**Proposed System Disadvantages**

- It is focused on post-query only.

## II. LITERTURE REVIEW

There is various works have been done as the research in many areas for Deep web search:

### *A. Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web*

The Deep web search is increasing by searchable databases online, in which information is hidden behind query. In this paper author proposes Meta Querier system for finding and integrating databases on the web. In this paper first proposes Meta Querier for Web-scale integration with its dynamic and ad-hoc nature. And second is this paper put the system architecture and methodology of their research work [2].

### *B. An interactive clustering-based approach to integrating source query interfaces on the deep Web*

There is lot of data sources increases but still there contents are accessible via query interfaces. Important thing of data source integration is we have to consider the integration of their query interfaces. Most important is we have eye the crucial step of the integration: accurately matching the interfaces. Now days query integration has more attention. Current approaches are not suitable for that first is they all model with flat schemes and second is they only consider 1:1 mapping over the interfaces and third is all the approaches work on black box techniques that if anything goes wrong then restart from scratch. This Paper presents clustering based approach to match query interface. Hierarchical behavior is catch by ordered trees. In this paper author proposes the human integrator back in the loop and various mapping parameters for resolution mapping [3].

*C. A hierarchical approach to model web query interfaces for web source integration*

In this paper consider the domain-independent common sense design rules, which are used to guide the creation of Web query interfaces. Transform query interfaces into schema trees by these rules. In this, Web query interface extraction algorithm proposed. This algorithm has HTML tokens and geometric layout of this token within web page. And using this layout tree structure is derived. And second tree is generated by field token. The Hierarchical representation of query interface is achieved by merging these two trees. In this ways they convert extraction problem into an integration problem [4].

*D. Deep Web Integration with VisQI*

This paper has VisQI -VISual Query interface Integration system used for Deep Web integration. VisQI has capability of first is transforming Web query interfaces into hierarchically structured representations second is classify representation into application domains and third is match the elements of different interfaces. Therefore VisQI is very good solution for hard challenges in building Deep Web integration systems. VisQI has portable components architecture that can be reused easily [5].

*E. Sampling Hidden Objects using Nearest-Neighbor Oracles*

There is various unknown set of objects embedded in the Euclidean plane and a nearest-neighbor oracle. In this how to calculate the set size and other properties of the objects is very important task. This is main task in this paper address this problem. They propose an efficient method that uses the Voronoi partitioning of the space by the objects and a nearest-neighbor oracle. Here main goal is to find number of interest objects in the hidden web/databases context. Nearest neighbor is located by a geographic location such as maps, local or store-locator APIs. They compare performance analysis with real world [9].

## III. PROPOSED APPROACH FRAMEWORK AND DESIGN

*A. Problem Definition:*

In this project we have proposed novel two- stage architecture based smart crawler that will efficiently search into deep hidden web resources. Hence harvesting better results than existing crawler where these hidden resources are not considered in the search. Because these resources are dynamic in nature, grow rapidly and forms huge volume of data.

*B. Proposed Methodology:*

In this venture Smart-Crawler contain novel two-stage design for a successful methodology for discovering information from the profound web. It has been demonstrated that above methodology accomplishes both wide degree for profound web interfaces and keeps up very effective slithering. Smart-Crawler is an engaged crawler comprises of two stages: site finding and adjusted in - site investigating.

In first stage Crawler will look contrarily for known profound sites i.e site finding. Smart-Crawler accomplishes more exact results by positioning gathered locales and centering the creeping on a given point. The in - site investigating stage utilizes versatile connection - positioning to look inside of a site and plan a connection tree so that to build the zone of hunt subsequently recovering better refined results. Shrewd Search is an engaged web hunt utilizing Page Rank Algorithm down productive, quick, all around organized indexed lists. Our proposed work accomplishes higher harvest rates than different crawlers.

At first Site finding stage will begin with seed destinations i.e locales in site database. On the off chance that number of unvisited URL's in database are not as much as limits. While creeping then Smart crawler will begin reverse seeking of the profound sites and nourish this information back to site database. Site wilderness will get landing page URL's from site database and the outcome is positioned by site ranker to sort them in need.

The positions given by webpage ranker can be enhanced by versatile website learner which logically learns highlight of profound pages found. To accomplish more applicable data we burrow for all the more profound website pages in our next stage i.e in-webpage investigating in which we dive profound into landing page content in the plan of seeking something important to hunt subject.

Once most applicable locales are discovered then we continue toward in-webpage investigating to perform profound web look. For this we experience all connections found profound inside the website pages. These connections are put away in connection wilderness and the pages are gotten and these are checked in the event that they are now gone to or not. if not encourage process occur. At that point connection is moreover set into hopeful wilderness so that to organize joins utilizing the connection ranker. The both phases of Smart-Crawler are entomb subject to one another to deliver joined and successful Search results. At the point when the crawler finds another site, the site's URL is

embedded into the Site Database. The connection ranker is exceptionally versatile to change in condition. It takes help of versatile connection learner so that to pursuit some unvisited any applicable URL's. Brilliant crawler additionally makes utilization of page rank calculation for burrowing profound pages. Page Rank is a numeric measure speaking to page significance on web which we have utilized as a part of the venture for Smart pursuit. Page Rank is a measure that without a doubt progresses in the direction of tally the number and nature of connections that site has. Consequently is utilized as a measure to choose significance of sites in this manner yielding great indexed lists.

*C. System Architecture.*

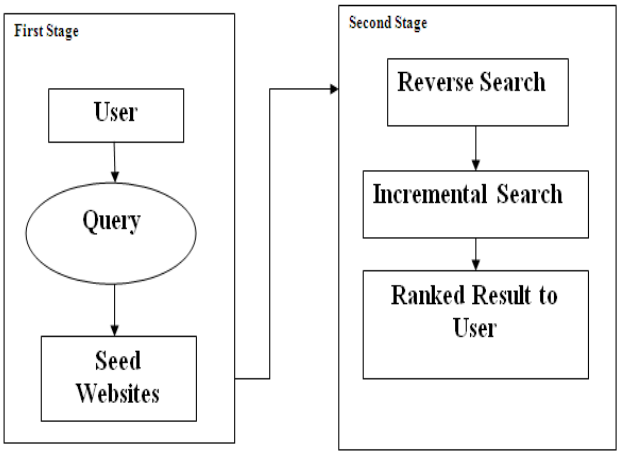


Figure1: System Architecture

*Explanation:*

The Figure 1 shows here the architecture for a novel two stage crawler which addresses the problem of searching deep web resources.

*Stage1: Site Locating*

At first Site finding stage will begin with seed locales i.e. destinations in site database. In the event that number of unvisited URL's in database are not exactly an edge. While creeping then Smart crawler will begin reverse seeking of the profound sites and encourage this information back to site database. Site outskirts will get landing page URL's from site database and the outcome is positioned by site ranker to sort them in need. The positions given by website ranker can be enhanced by versatile webpage learner which logically learns highlight of profound pages found.

*Stage2: In-site Exploring*

Once most applicable destinations are discovered then we continue toward in-webpage investigating to perform profound web seek. For this we experience all connections found profound inside the pages. These connections are put away in connection wilderness and the pages are gotten and these are checked in the event that they are as of now gone by or not. If not advance procedure happen. At that point connection are furthermore set into applicant outskirts so that to organize joins utilizing the connection ranker.

*D. Algorithm Used in Proposed System:*

**Algorithm 1: Reverse scanning for more locales.**

**Data: seed locales and gathered profound sites**
**Yield: pertinent locales**

1) while # of hopeful locales not exactly a limit do

2)/pick a profound site

3) website = getDeepWebSite(siteDatabase, seedSites)

4) resultP age = reverseSearch(site)

5) joins = extractLinks(resultP age)

6) for every connection in connections do

7) page = downloadPage(link)

8) pertinent = classify(page)

9) if pertinent then

10) relevantSites=extractUnvisitedSite(page)

11) Output relevantSites

12 end if condition

13 end for circle

14 end do while circle

**Algorithm 2: Incremental Site Prioritizing.**

**Data:** siteFrontier
**Yield**: searchable structures and out-of-site connections

1) HQueue=SiteFrontier.CreateQueue(HighPriority)

2) LQueue=SiteFrontier.CreateQueue(LowPriority)

3) while siteFrontier is not vacant do

4) if HQueue is vacant then

5) HQueue.addAll(LQueue)

6) LQueue.clear()

7) end if condition

8) site = HQueue.poll()

9) applicable = classifySite(site)

10) if applicable then

11) performInSiteExploring(site)

12) Output structures and OutOfSiteLinks

13) siteRanker.rank(OutOfSiteLinks)

14) if structures is not discharge then

15) HQueue.add (OutOfSiteLinks)

16) end if condition Line 14

17) else

18) LQueue.add(OutOfSiteLinks)

19) end else condition

20) end if condition Line10

21) end do while circle

**Calculation 3: Page Rank Algorithm:**

**Data:** Query Search results
**Yield**: Ranked Results

The first Page Rank calculation which was portrayed by Larry Page and Sergey Brin is given by

PR(A) = (1-d) + d(PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

## IV. MATHEMATICAL MODEL

**Let S, be a system such that,**
**S = {s, PR (A), PR (Ti), C (Ti), d}**
**Where,**
**S- Proposed System**
**s- Initial state at Time T=0.**
PR(A) – Page Rank of page A
PR(Ti) – Page Rank of pages Ti which link to page A
C(Ti) - Number of outbound links on page Ti
d - Damping factor which can be set between 0 and 1

**X- Input of System.**
**-We start from set of seed sites.**
**X be the input to system**

**Y- Output of System.**
**-Ranked results relevant to query.**

**T- Set of steps to be performed from Site Ranking to Link Ranking**

- **Site Ranking**

Smart Crawler perform Site ranking to prioritize deep sites of given query.
Here for site ranking we make use of two measure Site similarity and Site frequency.
Where, Site similarity defined by function Sim(U,Us) is used to find similarity between seed sites s and Known deep web sites.

**ST(s)=Sim(U,Us)**

Where, function Sim scores the similarity of the related feature between s and known deep web sites.
The function Sim(_) is computed as the cosine similarity between two vectors V1 and V2 as follows:
Sim(V1,V2)= V1.V2 / |V1| * |V2|

**Site Frequency:**

The site frequency measures the number of times a site appears in other sites. In particular, we consider the appearance in known deep sites to be more important than other sites. The site frequency is defined as:

**SF(s)**= E **Ii,**

Known site list
Where E stands for Summation and L represent Known site List
Where Ii = 1 if s appeared in known deep web sites, otherwise Ii = 0

- **Link Ranking**

Link ranking is done to prioritize the Links within the deep web sites and is defined as follows**,**

**LT(s)=Sim(P,Pl)**

Where function sim() calculate the similarity of the related feature between l and the known in-site links with forms. Finally, we use the link similarity for ranking different links.

## V. EXPERIMENTAL STUDIES

In this section we present the Module description, how it works, practical results and environment.

**1. MODULES**

**Info Seed Sites:**

•In this module we give the seed locales for info.

•Seed website are only beginning point from where our savvy crawler really start seek so that to investigate other profound pages.

**Site Locating:**

•We at first begin with seed locales i.e destinations in site database.

•If number of unvisited URL's in database are not as much as points of confinement. While crawling then Smart crawler will start reverse looking for of the significant destinations and reinforce this data back to site database.

•Site boondocks will bring presentation page URL's from site database and the result is situated by site ranker to sort them in need.

•The positions given by site ranker can be upgraded by flexible website page learner which ceaselessly learns highlight of significant webpage pages found.

•To accomplish more applicable data we burrow for all the more profound website pages in our next stage i.e in-webpage investigating in which we dive profound into landing page content in the plan of seeking something pertinent to inquiry point.

**In-Site Exploring:**

•Once most applicable locales are discovered then we continue toward in-webpage investigating to perform profound web seek.

•For this we encounter all associations discovered significant inside the site pages. These associations are secured in association wild and the pages are brought and these are checked if they are starting now passed by or not. if not encourage process happen.

•Then association is in addition set into contender edges so that to sort out joins using the association ranker.

•The both periods of Smart Crawler are dependent on each other to convey joined and suitable Search results. Right when the crawler finds another site, the site's URL is implanted into the Site Database.

•The join ranker is amazingly adaptable to change in condition. It takes help of adaptable association learner so that to interest some unvisited any relevant URL's.

**Positioning Sites utilizing Page Rank:**

•Page Rank is a numeric measure speaking to page significance on web which we have utilized as a part of the undertaking for Smart pursuit.

•Google decides Page rank relies on upon the votes been threw for a specific page. e.g. In the event that a site page is having its connection on numerous top positioned locales then naturally rank of this website page is additionally high.

•Page rank is the way Google characterizes significance of a website page. In any case, there are still numerous variables that add to positioning of site page in list items.

•Page Rank Notation - "PR".

## 2. HARDWARE AND SOFTWARE USED

- **Hardware Configuration**

- Processor -Pentium –IV
- Speed - 1.1 GHz
- RAM - 256 MB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Monitor - SVGA

- **Software Configuration**

- Operating System -Windows XP/7/8
- Programming Language - Java
- Tool - Netbeans.
- Server -Wamp Server

## 3. RESULTS OF PRACTICAL WORK

Results of work done are as shown in following output screen.
Figure 2 Shows Seed sites relevant to particular query. Figure 3 shows how the rank sites it will get displayed according to their rank value. Figure 4 shows Graphical Representation of Existing System Vs Proposed System

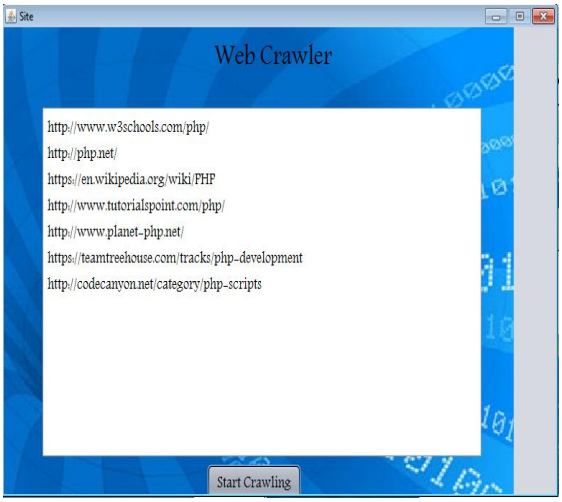


Figure 2: Seed sites relevant to particular query

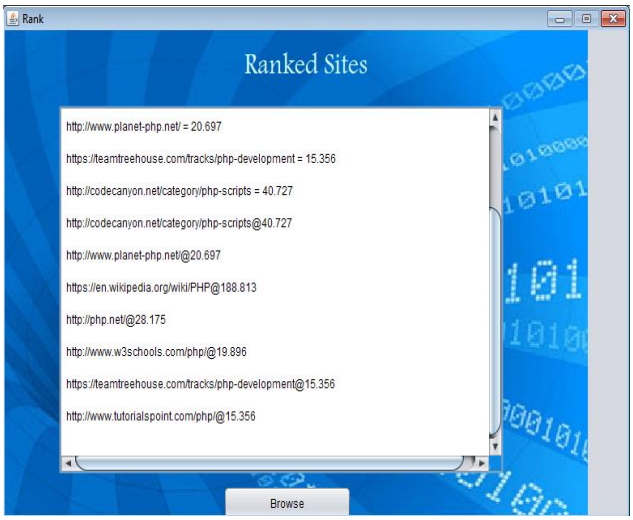


Figure 3: Displaying the ranked sites according to rank value

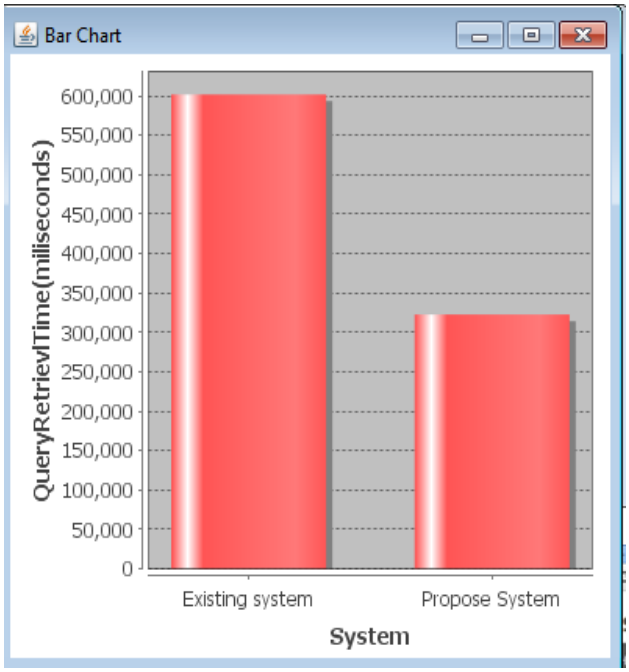


Figure 4: Graphical Representation of Exiting verses Proposed System

So in this way our Proposed System is much more efficient as compared to the Existing System because for our Proposed System it will took very less time to retrieve the Query information

## VI. Complexity Analysis

In this segment we arrive to examine the Performance of our framework regarding time. Time intricacy of Algorithm gives us measure of time required for execution of capacity for a standard info .Usually we utilize Bog O documentation to speak to the most exceedingly bad time many-sided quality i.e Maximum time required by Algorithm to perform a Function for given information.

This will altogether depend on upon data structure used for executing the computation. Case in point the data structure used to realize the boondocks need line in the Best-First computation. The central operations of embeddings and ousting joins from the line are most gainfully reinforced by a store, yielding O(log(MAX BUFFER)) multifaceted nature. In any case it is moreover imperative to check if an as of late isolated URL is starting now in the wild; this has straight many-sided nature in a heap, and ought to be done each association in an as of late crawled page. Then again, using a familiar bunch or hash table, this checking operation is unassuming (O(1)) yet embeddings and clearing joins requires an excessive sorting operation (O(MAX BUFFER • log(MAX BUFFER))) once per page crawled.

## VII. Conlusion

We have proposed a successful methodology for discovering information from the profound web. It has been exhibited that above philosophy performs both wide augmentations for significant web interfaces and keeps up extraordinarily profitable crawling. Keen Crawler is a connected with crawler involves two stages: site finding and balanced in - site researching.

In first stage Crawler will scan conversely for known profound sites i.e site finding. Keen Crawler achieves more correct results by situating accumulated destinations and focusing the inching on a given point. The in - site page researching stage uses flexible association - situating to look for within a site and design an association tree for wiping out slant toward particular registries of a webpage for more broad extent of web inventories. Shrewd Search is an engaged web hunt utilizing Page Rank Algorithm down proficient, quick, very much organized indexed lists. Our proposed work achieves higher harvest rates than various crawlers.

## VIII. Future Scope

In future work, we have a course of action to mix the pre-question approach and post-request approach together to

perceive significant web resources so that to help increase the capability and accuracy of our web searcher.

## IX. ACKNOWLEDGMENT

We would like to thank my Project Guide Dr. Archana Lomte for her guidance and for giving me her valuable time. She helped in a broad range of issues from giving me direction, helping to find the solutions and giving me her valuable time. Also we would like to thank Dr. G. M. Bhandari, Head of Department of Computer Engineering to encourage me to go ahead and for continuous guidance.

## REFERENCES

[1]Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin, Smart-Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces.IEEE Transaction on service computing vol 99 Year 2015

[2] 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.

[3] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pages 95–106. ACM, 2004.

[4] Eduard C. Dragut, Thomas Kabisch, Clement Yu, and Ulf Leser. A hierarchical approach to model web query interfaces for web source integration. Proc. VLDB Endow, 2(1):325–336, August 2009.

[5] 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.

[6] Jayant Madhavan, David Ko, Łucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google's deep web crawl. Proceedings of the VLDB Endowment, 1(2):1241–1252, 2008.

[7] 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.

[8] Panagiotis G Ipeirotis and Luis Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. In Proceedings of the 28th international conference on Very Large Data Bases, pages 394–405. VLDB Endowment, 2002.

[9] Nilesh Dalvi, Ravi Kumar, Ashwin Machanavajjhala, and Vib-hor Rastogi. Sampling hidden objects using nearest-neighbor oracles. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1325– 1333. ACM, 2011.

[10] Olston Christopher and Najork Marc. Web crawling. Foundations and Trends in Information Retrieval, 4(3):175–246, 2010.

[11] Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin Dong, David Ko, Cong Yu, and Alon Halevy. Web-scale data integration: You can only afford to pay as you go. In Proceedings of CIDR, pages 342–350, 2007.

**Ms. Sneha Avinash Ghumatkar** completed BE in Information Technology from Dnyanganga College of Engineering Narhe, Pune in 2011. Now she is pursuing her Master of Engineering in Computer Engineering from JSPM'S Bhivrabai Sawant Institute of Technology & Research Wagholi, Pune, Savitaribai Phule Pune University.

**Dr. Archana C. Lomte** currently working as Assistant Professor and PG Coordinator of Computer Engineering Department, At, JSPM'S Bhivrabai Sawant Institute of Technology & Research Wagholi, Pune, She had pursued her Ph.D from JJTU, Rajastan, 2016, Before that She had Completed her B.Tech(Computer Engg) From SNDT University Mumbai, 2004, Also Completed M.E.(Computer Engg) From PICT, Pune, 2010, She has around 10 years of Teaching Experience.