ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Efficient Hidden Web Harvesting Using Two Stage Adaptive Crawler

Hojage Swati B, Prof. Bharathi H.N.

Department of Computer Engineering, K.J. Somaiya College of Engineering, Vidyavihar, Mumbai,India

ABSTRACT: Traditional search engines deal with the Surface web which is a set of web pages directly accessible through hyperlinks and ignores a large part of the web called hidden web which is a great amount of valuable information of online database which is "hidden" behind the query forms. To access to those information the crawler have to fill the forms with a valid data.Recently, there has been increased interest in the retrieval and integration of hidden-Web interfaces or deep web data as it grows rapidly. Due to the dynamic nature of the Web, where data sources are constantly changing, it is difficult or challenging to automatically discover these interfaces. Hence tend to propose a two stage adaptive crawler for efficiently harvesting the deep web or hidden web interfaces which cannot be indexed by search engines. To visit large number of pages, it takes more time. So, with the help of search engine the two stage adaptive crawler perform site-based searching for centre pages. This is first stage of propose crawler. Then adaptive link ranking is used for fast searching in in-site. This is the second stage of adaptive crawler.

KEYWORDS: Hidden web, adaptive learning, ranking, feature selection.

I. INTRODUCTION

The deep web or hidden web is the part of internet that is inaccessible to conventional search engines and to most users. From analysis, hidden web has data in TB but it's one fourth is also not in web surface. Many data would be stored as relational data or structured data. hidden web is about 500 times larger than surface web. All data including in deep web contains important information in private databases. But these data is not index by search engines. So it is not much viewed by users. There is need for exploring this type of data accurately and quickly. Crawler can search databases of deep web and explore all data. The task of exploring databases of deep web is bit some challenging work because no search engines register deep web data. Data is changing constantly as new sources are added and old sources are removed and also it is distributed sparely. To address this problem Generic crawlers and Focused crawler were used in which generic crawler such as Form Focused crawler[3] fetch all searchable forms but doesn't fetch the data on a single topic but Focused crawler does this by fetching the data by automatically searching the databases on specific topic.FFC is designed with link, page and form classifiers for focused crawling of web forms.

1.1 Deep website crawling

There are many Search Engine are used over the WWW but the problem with these general search engine is that they are best for surface web searching but not too good for deep web searching in which what an end user expecting that maximum relevant documents must retrieved with his query But as the space on WWW is increasing it contains a vast amount of data, of an valuable information and these information cannot access properly by web indices in web search engine (e.g. Google, Baidu) then to overcome these problem there is need of efficient harvesting which accurately and quickly explore the deep web, it is challenging to locate a deep web database because they are not register with any search engine. To address this problem previously working on two types of crawler first is generic crawler and second is focused crawler, Focused crawler search automatically on-line database from to search engine, generic crawler is hidden or adaptive crawler [1].

However, search engines have some limitations as they operate on fixed algorithms, often leading to irrelevant results because the search engine is sometimes not able to contextualize the query. Also, search engine bots only

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

crawl static web pages, whereas a majority of the information on the net is stored in databases, which the spiders are not able to crawl. Thus, the search results get miss out on the data in several databases, such as those of universities and government organizations, among others. The most logical question then is, if Google can't find the data, where exactly is it and why can't it be crawled? A database contains information stored in tables that are created by programs such as Access, SQL or Oracle. This data can only be retrieved by posting a query. The query, when executed, searches the database to come up with the result that has been specified. This is very different from searching static web pages that can be accessed directly by crawlers.[2]. So to harvest the deep web and to provide the answer of user query with minimum effort proposing system will give the adaptive crawling concept which is based on two stage crawling.

II. RELATED WORK

2.1 Overview of hidden web

The hidden web provides access to huge and rapidly growing data repositories on the web. Some authors have obtained approximations to its huge size: In 2001, an initial study by Bergman indicated the size of the data in the hidden web to be approximately 500 times the size of the data in the Surface Web which included as many as 43,000-96,000 web sites offering access to 7500 terabytes of data [4]. Later, in 2004, Chang et.al. Used a random IP sampling approach to measure the hidden web content in online databases and revealed that most of the data in such databases is structured [5]. Further in 2007, Ben He et.al. by analyzing the percentage overlap between the most commonly used search engines such as Yahoo!, Google and MSN discovered the number of such sites to 236,000- 377,000 with only 37% of the available content being indexed by these search engines [6]. Thus, according to experts, the hidden web forms the largest growing category of new information on the internet and comprises of: ¬ nearly 550 billion documents ¬ content high relevant to every information need, market and domain ¬ Up to 2,000 time's greater content than that of the Surface Web. 95% publicly accessible information not subject to fees or subscription.

The contents of the hidden Web can only be accessed by filling out web forms, such as the ones in Figure 2.1.1 , which are then submitted as queries to the online database behind the form. The hidden web covers several topic domains, such as government, education, entertainment, business, health, news, and sports. There are thousands of online databases for each of those domains – most of them containing structured information [5]. Exposing the contents of an online database can be achieved by designing wrappers, i.e., programs that extract data from a specific Web site. However, since wrappers are specific for each site, this solution is not feasible when dealing with thousands of hidden Web sites. Thus, a more scalable approach is to use a hidden web crawler to automatically identify and retrieve data from online databases. Hidden web crawling has many applications. The discovered content can be indexed by generic search engines, such as Google, Bing, and Yahoo!. Furthermore, it can be used to create vertical search engines, which focus on a specific segment such as real estate, online auction, books, airline tickets ,etc. The identification of the fields is reasonably straightforward and can be achieved by parsing the HTML code of the page containing the form looking for specific tags such as input and select. The challenge is Web Form Filling (WFF), i.e., how to automatically fill the fields using suitable values in order to retrieve meaningful data. This is a critical step since filling a form with unsuitable values will result in blank or error pages representing a waste of resources[13].

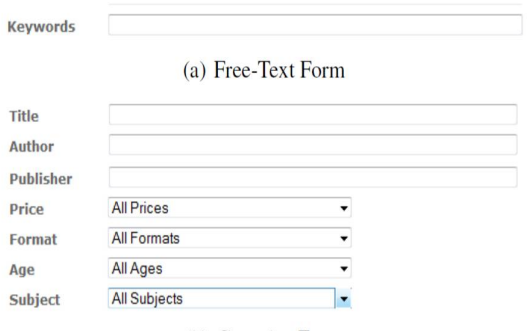

fig 2.1.1:(a) Free text form & (b) complex form[13]..

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Figure 2.1.2 shows a fragment of the HTML code for the web form shown in Figure 2.1.1(b). An HTML form is defined within a <form> and a </form> tag . Form fields can be text boxes, selection lists, checkboxes, radio buttons, or submit buttons. Selection lists, radio buttons, and checkboxes show a list of options to the user. On the other hand, a text box field does not contain options, so the values need to be discovered somehow. More concisely, fields can be grouped into two types: fields with a finite domain such as selection lists and fields with an infinite domain, such as text fields, in which a user can type any value[12].

```
1<form method="get" action="/results.html">
2     <span>Title</span>
3     <input type="text" name="title" />
4     <span>Author </span>
5     <input type="text" name="authorName" />
6     <span>Publisher</span>
7     <input type="text" name="publisher" />
8     <label for="price">Price</label>
9       <select name="prc" id="prc">
10        <option value="all">All Prices</option>
11        <option value="1">under $10</option>
12        <option value="2">$10-$25</option>
13        ...........
14      </select>
15    <label for="select">Format</label>
16      <select name="fmt" id="fmt">
17        <option value="all">All Formats</option>
18        <option value="0">Hardcover</option>
19        <option value="1">Paperback</option>
20        ...........
21      </select>
22      ...............
23</form>
```

fig 2.1.2 Fragment of HTML code[13].

2.2 Deep web crawler framework

The basic actions of a deep web crawler are similar to those of other traditional crawlers [15].In Figure 1 the flowchart on the left indicates the typical crawler loop, consisting of URL selection, page retrieval, and page processing to extract links. Note that traditional crawlers do not distinguish between pages with and without forms Specifically, deep web crawler performs the following sequence of actions for each form on a page:

Step 1 Parse and process the form to build an internal representation, based on the model outlined in Section

Step 2 Generate the best (untried) value assignment and submit a completed form using that assignment.(Value assignment and submission)

Step 3 Analyze the response page to check if the submission yielded valid search results or if there were no matches. This feedback could be used to tune the value assignments

Step 4 If the response page contains hypertext links, these are followed immediately (except for links that have already been visited or added to the queue) and recursively, to some pre-specified depth. However, for ease of implementation, in deep web crawler, user chose to navigate the response pages immediately and that too, only up to a depth of 1. Steps 2, 3, and 4 are executed repeatedly, using different value assignments during each iteration [16].

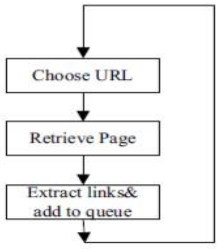

Fig:2.2.1 Traditional crawler loop[16]

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

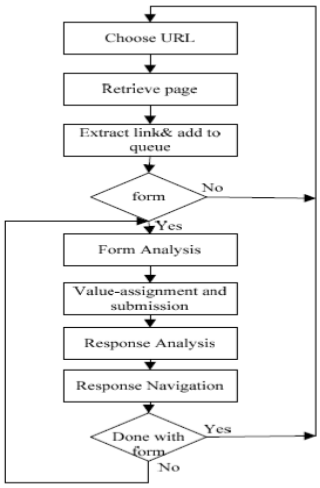

Fig2.2.2 Deep web crawler loop[16].

2.3 Searching for hidden database

The author in[3]proposed a crawling strategy to automatically locate hidden web databases which also aims to achieve a broad search by avoiding need to crawl large number of irrelevant pages. The proposed strategy does that by focusing the crawl on a given topic; by choosing links to follow within a topic that are more likely to lead to pages that contain forms. Proposed system has used page classifier to guide the crawler and focus the search on pages that belong to a specific topic . The crawler uses two classifiers to guide its search: the page and the link classifiers. A third classifier, the form classifier, is used to filter out useless forms. The page classifier is trained to classify pages as belonging to topics(e.g arts ,movies,jobs etc). It uses the same strategy as the best-first crawler. Once the crawler retrieves a page P, if P is classified as being on-topic, forms and links are extracted from it. A form is added to the Form Database if the form classifier decides it is a searchable form, and if it is not already present in the Form Database. The link classifier is trained to identify links that are likely to lead to pages that contain searchable form interfaces.

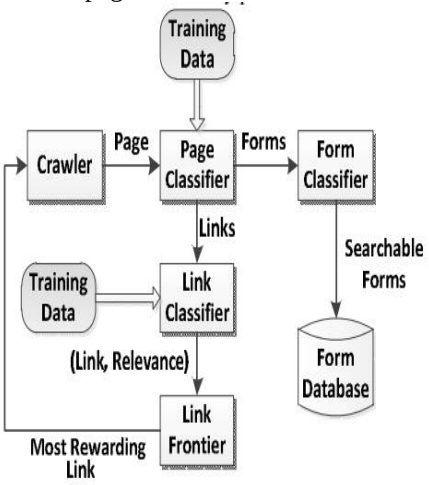

Figure 2.3.1 Form Focused Crawler[3]

The crawler combines the use of a page classifier and a link classifier that have been trained for focusing its crawl on a particular topic by taking into account the contents of pages and patterns in & around the hyperlinks paths to a web page. The frontier manager is another major component of the FFC framework and is used to select the next target link for crawling based on their reward values decided by the current status of the crawler and the priority of the link in the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

current crawling step. The FFC also uses a form classifier to filter out useless forms. If a form is found searchable by the form classifier, it is added to the form database if not already present in it.
Meta Querier [7]is a system that enables large-scale integration of hidden-Web data. It consists of several components that address different aspects of the integration. One of these components is a crawler for locating online databases, the database crawler. Unlike proposing approach, the crawler neither focuses the search on a topic, nor does it attempt to select the most promising links to follow. Instead, it uses as seeds for the crawl the IP addresses of valid Web servers; then, from the root pages of these servers, it crawls up to a fixed depth using a breadth-first search.

III. PROPOSED WORK

3.1 Two stage System architecture

Different from the crawling techniques and tools as discussed in previous section the propose system is a two stage crawler which is domain specific for locating relevant deep web site content sources. This adaptive crawler not only classifies sites in first stage to filter out irrelevant sites but also categorizes searchable forms in another stage. Instead of just classifying links as relevant or not, this focused crawler is going to rank the sites through site ranker and then prioritize the extracted links.

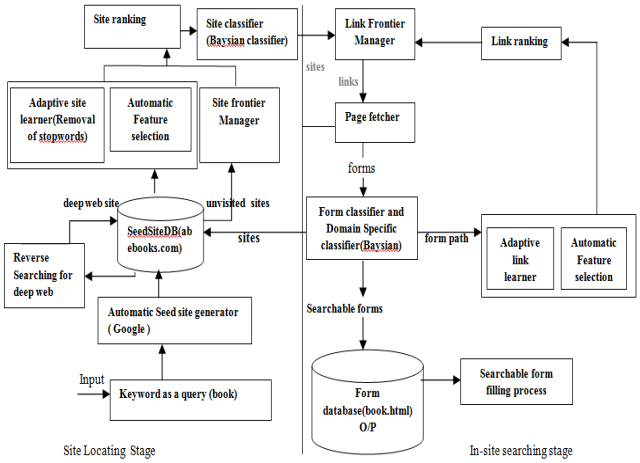

figure 3.1.1 Two stage crawler architecture

.

3.2 Overview of proposed system

3.2.1 Stage 1:Site Locating

The site locating stage starts with seed set of sites in the site database. Crawler starts crawling by following URLs from chosen seed sites to search other pages and domains. When the number of unvisited URLs in database are less than some threshold during crawling then the propose crawler does "reverse searching" of deep web sites by choosing center pages which are highly ranked by search engines which have links to other domains and store this new entries back to site database which lacks in existing crawlers.
Site locating stage finds relevant sites for a given topic which consist of three blocks site collecting, site ranking and site classification.

3.2.1.1 Site Collecting

Generally most of traditional crawlers follows all newly found links. But this proposing approach tries to minimize visited URLs and at the same time it maximizes number of deep web sites. for achieving this links in downloaded pages are not enough because a web site contains small number of links to other sites. To address this problem this adaptive crawler is going to do "reverse searching" which will find more links with the help of search engines by finding centre page of unvisited sites.
Reverse Searching: One can randomly pick a known website or a seed site and use general search engines facility for find centre pages which are ranked by that search engine. The result page will extract the links. Using following heuristic rules downloaded pages are analyzed whether the links are relevant or not.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

- If the page contains related searchable forms it is relevant.
- If number of deep web sites in page is larger or near to threshold then page is relevant.

Finally user will get relevant links and site frontier will be with enough sites for further crawling.

Algorithm 1: Reverse searching for more sites.
input : seed sites and harvested deep websites
output: relevant sites
1 while # of candidate sites less than a threshold do
2 // pick a deep website
3 site = get Deep web site(seed site)
4 result Page = reverse Search(site)
5 links = extract Links(result page)
6 for each link in links do
7 page = download Page(link)
8 relevant = classify(page)
9 if relevant then
10 relevant Sites =extract Unvisited Site(page)
11 Output relevant Sites
12 end [1].

3.2.1.2 Site ranking

Site Frontier fetches homepage URLs from the site database, which are ranked by Site Ranker to prioritize highly relevant sites. The Site Ranker is improved during crawling by an Adaptive Site Learner, which adaptively learns from features of deep-web sites (web sites containing one or more searchable forms) found. Site Ranker assigns a score for each unvisited site that corresponds to its relevance to the already discovered deep web sites. Two stage crawler ranks site URLs to prioritize deep sites of a given topic. Site similarity and site frequency, are considered for ranking. Site similarity measures the topic similarity between a new site and known deep web sites. Site frequency is the frequency of a site to appear in other sites.

The feature space of deep web sites (FSS) is defined as:

$$FSS = f(U; A; T) \quad (1)$$

where U, A, T are vectors corresponding to the feature context of URL, anchor, and text around URL of the deep web sites.

Given the homepage URL of a new site s =(U_s, A_s, T_s) the site similarity can be defined as

$$ST(s) = Sim(U,U_s) + Sim(A,A_s) + Sim(T,T_s) \quad (2)$$

here function Sim() is computed as is computed as the cosine similarity between two vectors V_1 and V_2 [10].

$$Sim(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| \times |V_2|}.$$

The site frequency measures the number of times a site appears in other sites .

3.2.1.3 Site classification

After ranking Site Classifier categorizes the site as topic relevant or irrelevant for a focused crawl, which is similar to page classifiers in FFC[3] in which they used naive Bayes classifier, to build page classifier. If a site is classified as topic relevant, a site crawling process is launched. Otherwise, the site is ignored and a new site is picked from the frontier. In Smart crawler the system determine the topical relevance of a site based on the contents of its homepage. When a new site comes, the homepage content of the site is extracted and parsed by removing stop words and stemming.

3.2.2 Stage 2: In-site Searching

Once a site is regarded as topic relevant, in-site -searching is performed to find searchable forms. The goals are to quickly harvest searchable forms and to cover web directories of the site as much as possible. To achieve these goals, in-site searching adopts two crawling strategies for high efficiency and coverage. Links within a site are prioritized with Link Ranker which is similar to site ranker and Form Classifier classifies searchable forms.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

3.2.2.1 Link Ranking

Link ranker prioritizes links so that crawler can quickly discover searchable forms. For prioritizing links of a site, the link similarity is computed similarly to the site similarity but link frequency is not considered in link ranking.

3.2.2.2 Form classification

Since the goal is to find hidden-web databases, user need to filter out non-searchable forms, e.g., forms for login, discussion groups interfaces, mailing list subscriptions, purchase forms, Web-based email forms. This proposing ur approach consists of two classifiers, a searchable form classifier (SFC) and a domain-specific form classifier (DSFC). SFC is a domain-independent classifier to filter out non-searchable forms by using the structure feature of forms..

3.3 Feature selection and adaptive learning

3.3.1 Online construction of feature space

This section first discusses the online feature construction of feature space and then adaptive learning process. In Smart crawler, patterns of links to relevant sites and searchable forms are learned online to build both site and link rankers.. The feature space of deep web sites (FSS) is defined as:

$$FSS = \{U; A; T\}$$

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:

$$FSL = \{P; A; T\}$$

where A and T are the same as defined in FSS and P is the vector related to the path of the URL.

3.3.2 Adaptive Learning

Adaptive crawler schema in this system adaptively learns and update information collected successfully during crawling. Periodically, FSS and FSL are adaptively updated to reflect new patterns found during crawling. As a result, site ranker and link ranker are updated. 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. The adaptive link learner, in contrast, uses features of paths that are gathered during the crawl. The adaptive link learner is invoked periodically, when the learning threshold is reached .

Algorithm 2: Adaptive link learner
1: if learning threshold reached then
2: paths = collect Paths(relevant Forms, length)
{Collect paths of a given length to pages that contain relevant
forms.}
3: features = Feature Selector(paths)
{Select the features from the neighborhood of links in the
paths.}
4: link classifier = create classifier(features, paths)
{Create new link classifier.}
5: update Frontier(link classifier)
{Re-rank links in the frontier using the new link classifier.}
6: end if [1].

IV. EVALUATION

System performed an extensive performance evaluation of crawling framework over real web data in few domains. Goal of this system includes:

1. Improve Efficiency of crawling
2. Achieves wide coverage for deep web interfaces and analyzing effectiveness of site collecting.
3. Achieves the performance of adaptive learning process.
4. Automatic searchable form filling to discover hidden web. Two stage crawler is implemented using .Net framework.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

4.1 Experimental setup and results

Two stage crawler is implemented using .Net framework.
Domains for our experiment are as follows:

Table: Domains for experiment

Domain	Description
Book	Book search
Job	Job Search
Movie	Movie titles search
Hotel	Hotel search

Table: Results of experiment

Domain name	Relevance score(%)	No. of pages crawled	No. of forms found	Indexing Time (min)
Book	76%	2995	248	49
Job	83%	2020	141	35
Movie	79%	2101	150	36
Hotel	88%	2282	33	38

V. CONCLUSION AND FUTURE SCOPE

In this system an effective harvesting framework for deep-web interfaces has been proposed. Two stage deep web crawler is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring. The two stage crawler performs site-based locating by reversely searching the known deep web sites for centre pages, which can effectively find many data sources for sparse domains. By ranking collected sites and by focusing the crawling on a topic, propose crawler achieves more accurate results. The in-site exploring stage uses adaptive link-ranking to search within a site.

Future Scope: As hidden web also contains unstructured data[text, images, videos] so in future work system can crawl deep web videos and also try to capture technically limited deep web contents which requires CAPTCHA technology.

REFERENCES

[1] Feng Zhao, J. Z. (2015). "Smart Crawler:Two stage Crawler For Efficiently Harvesting Deep-Web Interface". IEEE Transactions on Service Computing Volume:pp Year :2015.
[2] http://www.livemint.com/Leisure/2Y84mzsoI1AH64A5LKKLhN/Crawling-the-deep-web.html
[3] Luciano Barbosa and Juliana Freire. "Searching for hidden-web databases". In WebDB, pages 1–6, 2005
.[4] Michael Bergman, "The deep Web: surfacing hidden value". In the Journal Of Electronic Publishing 7(1) (2001).
[5] Moumie Soulemane"Crawling the Hidden Web: An Approach to Dynamic Web Indexing" International Journal of Computer Applications (0975 – 8887) Volume 55– No.1, October 2012.
[6] K.C. Chang, B. He, M.Patel, Z.Zhang :" Structured Databases on the Web: Observations and Implications". SIGMOD Record, 33(3). 2004.
[7] B. He, M.Patel, Z.Zhang, K.C. Chang: "Accessing the Deep Web: A survey". Communications of the ACM, 50(5):95–101, 2007
[8] . K. C.-C. Chang, B. He, and Z. Zhang. "Toward Large-Scale Integration: Building a MetaQuerier over Databases on the Web". In Proc. of CIDR, pages 44–55, 2005.
[9] Infomine. UC Riverside library. http://lib-www.ucr.edu/,2014.
[10] Clusty's searchable database dirctory. http://www.clusty.com/, 2009.
[11] Soumen Chakrabarti, Martin Van den Berg, and Byron Dom."Focused crawling: a new approach to topic-specific web resource discovery". Computer Networks, 31(11):1623–1640, 1999..
[12] https://en.wikipedia.org/wiki/Cosine_similarity

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

[13] Dumais Susan and Chen Hao. Hierarchical classification of Web content. In Proceedings of the 23rd Annual International ACM SIGIR conference on Research and Development in Information Retrieval, pages 256–263, Athens Greece, 2000.
[14] Gustavo Zanini Kantorski Viviane Pereira Moreira Carlos Alberto Heuser." Automatic Filling of Hidden Web Forms: A Survey".
[15].https://en.wikipedia.org/wiki/Tf%E2%80%93idf
[16] CHAKRABART J S , MARTIN V D B, BYRON D." Focused crawling: A new approach to topic specific web resource discovery."Proceedings of the Eighth International World-Wide Web Conference, 1999
[17] Xiang Peisu1, Tian Ke2, Huang Qinzhen1" A Framework of Deep Web Crawler".