Hidden Web Data Extraction Tools

Babita Ahuja
Manav Rachna College of Engineering
Faridabad, India

Anuradha, Ph.D
YMCA University
Faridabad, India

Ashish Ahuja
Computer Science Corporation
Noida, India

ABSTRACT
This Hidden web forms 99% of the total web. The hidden web contains the high quality content and has a broad coverage. So hidden web has always remained like a golden apple in the eyes of the researcher. A lot of research has been carried out in this area. The different tools have been created by the researchers to make the hidden web float on the surface of WWW. The different kind of crawlers and the search engines have been developed which focuses on the hidden web. So in this paper we discuss the various kinds of web crawlers and search engines which throw some light on the hidden web.

Keywords
Hidden web; surface web; WWW; crawlers; search engines

1. INTRODUCTION
Initially the WWW was small and browsing i.e. following the hyperlinks was an adequate method for locating relevant information. But when WWW grew in size and diversity, the process of following a hypertext trail of links created by other Web users became difficult. So special tools were created that could satisfy the users information needs. Two search tools [1] were created which assisted the users to locate the information on the WWW. The first tool was "Web Directories". They provided a context-based framework for structured browsing like Archie, Gopher, Yahoo!, LookSmart, and the Open Directory Project (ODP). The second tool was "Search Engine". Search Engines allowed searching for specific keywords or phrases like google and altavista. Web directories are similar to a table of contents in a book; search engines are more similar to an index. Web directory like the table of contents allows the reader to quickly turn to interesting sections of a book by examining the titles of chapters and subchapters. Search Engine like a book's index offers a much finer level of granularity, providing explicit pointers to specific keywords or phrases regardless of where they appear in the book. The www is composed of two components.

- Surface Web
- Hidden Web

Surface Web is that part of the WWW that is index able by the traditional search engines. Surface Web forms 1% of the total WWW. Hidden Web is also known as invisible web or deep web. The hidden web forms 99% of the total WWW as shown in Figure 1. This content is not easily located with the information-seeking tools i.e. search engines which is used by most Web users. The traditional search engine's web crawler is not sufficed to crawl the hidden web. But in this era of digital tsunami of information on the web, everyone is completely dependent on the WWW for information retrieval. Since surface web forms only 1% of the WWW.

Fig1: Hidden Web and Surface Web

So only 1% of WWW is indexed by search engines and is accessible to the users. 99% of the WWW is composed of the Hidden Web [2]. So it is a very large part of WWW. The data residing in the hidden web is of high quality. The data residing in the hidden web is of high quality. Users are missing the Hidden Web so it means missing plenty of important and amazing resources of the web.

1.1 TYPES OF INVISIBILITY IN WWW
There are certain kinds of material on the web which cannot be accessed via the internet as they are not indexed by the search engines. WWW has four types [1] of hidden web or Invisible content as shown in Figure 1. The four types of invisibility are:

• The Opaque Web

• The Secret Web

• The Proprietary Web

• The Proper Invisible Web

Many researchers have contributed in the task of fetching the data from these deep web resources and pouring it on the user's screen. Their contribution can be categorized in the following given domains.

- Hidden Web Crawlers
- Hidden Web Search Engines
- Wrapper Generation

So in this paper we are going to illustrate the work carried out by different researchers in different domains.

2. HIDDEN WEB CRAWLERS
All The different web crawlers are created by the researchers to put the deep web on the surface. In this section we are going to discuss the different hidden web crawlers, their advantages and their disadvantages.

2.1 Deepbot

Manuel Alvarez developed a hidden-web crawler called Deepbot[3] to access Deep web. Deepbot crawl both the client-side hidden web and server -side hidden web. During the crawl of client-side hidden web deepbot take care of client side script such as JavaScript, VbScript and session maintaining mechanism. Traditional crawlers are not able to handle client-side scripts and session maintaining mechanism. Deepbot has a "mini-web browser" to handle client-side scripts and session maintaining mechanism. During the crawl of server-side hidden web deepbot maintains the domain definitions which help in fetching data behind query interfaces which traditional crawlers fails to do. Deepbot has the following components:

2.1.1 Route Manager Component

It is responsible for maintaining the master list of all routes that needs to be accessed. There is a pool of crawlers. All the crawlers from the pool are able to access this master list of URL. Route manager records the URL as well as the session information which helps in fetching pages from deep web

2.1.2 Configuration Manager Component

This component holds the information that is needed by the crawlers when they start crawling. The information about the seed URL's, the depth of crawl, different download handlers for downloading different kind of files (images, Pdf, MS Word, html etc.), regular expression that helps in discarding the URL's which are not worth download and selecting the important URL's which are worth download.

2.1.3 Download Manager Component

The crawlers start downloading the document using both the mini web browser and domain definitions. Download Manager Components selects the appropriate handler depending the type of download document (images, Pdf, MS Word, html etc.)

2.1.4 Content Manager Component

The thorough scrutiny of downloaded documents is done by this component Content Manager. Content Manager employs a pair of filters to accomplish this task. The first filter is used to decide the relevance and the quality of web page. If page is of poor quality it is discarded else it is stored and indexed. The second filter has two filters embedded in it. The first filter called Obtain Links retrieve the URL from the downloaded document. The second filter called form analyzer filter analyses every form and determines the relevancy for any of the preconfigured domain definitions.

2.1.5 Data Repository

It stores all the pages from both the surface web and the deep web.

2.1.6 Indexer

All the downloaded documents stored in the data repository are indexed by this module. The indexing is done for fast access of the downloaded documents.

2.1.7 Searcher

The generic searcher and the ActiveX searcher are used to fetch the documents of user interest.

Advantages of Deepbot

- It fetches the data behind the web query interfaces.
- It handles both the client-side deep web and the server-side deep web.

Disadvantages of Deepbot

- Most web query interfaces can have multiple input fields and have multiple input values for those fields. So filling form accurately is a great issue. There will be many permutation and combinations for these Web Interfaces.
- Millions of queries need to be submitted to get the Hidden Data
- Most websites containing hidden data update very frequently. So crawling needs to be done frequently.
- Mass storage will be needed to store web pages from the surface web and hidden web.

2.2 HiWE

Sriram has created a Hidden web crawler called HiWE[4]. HiWE stands for Hidden Web Exposer. HiWE is a task-specific hidden web crawler. HiWE extracts the data hidden behind the web query interfaces. HiWE automatically process, analyze and submit the forms. The main components of HiWE are:

2.2.1 URL List

The crawlers crawls the URL's. All the URL's that the crawler has crawled are stored in this URL List.

2.2.2 Crawl Manager

Crawl Manager has got the full responsibility of the crawling process. The crawl manager is provided with the set of the websites that the crawler has to crawl.

2.2.3 Parser

The parser extract the URL's from the pages that the crawler has crawled. After extracting these URL's these URL's are then saved in the URL's list.

2.2.4 Form Analyzer

The form analyzer analyses the web form page and extract all the information. Using the information extracted it creates internal representation of the form where it stores all the elements of the form along the submission information and metadata of the web pages (F= ({E1, E2…En}, S, M)). It uses the Layout Based Information Extraction Technique to extract the semantic information from the search forms.

2.2.5 Form Processor

Form processor automatically fills the web query interfaces and then automatically submits the web forms. Form processor accesses the LVS table via LVS Manager to auto fill the web query interfaces.

2.2.6 Response Analyzer

After the form submission by form processor the response pages are received by the response analysis module. It stores the pages in the crawler's repository. It also recognizes the pages containing the search results and the pages containing the error messages. This analysis of result pages helps in later assignment of values to the form elements.

2.2.7 Label Value Set(LVS)

The values that are used in HiWE to fill the query interface forms are stored in LVS table. The table consists of Label and value set. The value set has a value and a number that indicates the effectiveness of that value.

Advantages of HiWE

- It fetches the data behind the web query interfaces.

- It uses the LITE technique to extract the semantic of the web forms.

Disadvantages of HiWE

- HiWE is not able to recognize and respond to dependencies between form elements.
- HiWE lacks support for partially filling out forms.
- Human Assistance is needed. So it is not fully automatic.
- It is task Specific.
- Most web query interfaces can have multiple input fields and have multiple input values for those fields. So filling form accurately is a great issue. There will be many permutation and combinations for these Web Interfaces.
- Millions of queries need to be submitted to get the Hidden Data
- Most websites containing hidden data update very frequently. So crawling needs to be done frequently.
- Mass storage will be needed to store web pages from the surface web and hidden web. Please do not revise any of the current designations.

2.3 Incremental Web Crawler

Dr. Komal Kumar Bhatia proposed an incremental web crawler [5]. The incremental web crawler continuously refreshes the Web Repository. It updates the repository of search engine by re-crawling the web pages that are updated more frequently. The incremental web crawler uses a mechanism for adjusting the time period between two successive revisit of the crawler based on the probability of the web page. The Incremental Web Crawler has the following components:

2.3.1 Domain Specific Hidden web crawler (DSHWC)

DSHWC creates the unified query interfaces from downloaded hidden web pages, auto fills that interface and auto submits the query. The response pages of the query are then downloaded and stored in hidden web pages repository along with their URL's.

2.3.2 URL Extractor

It extracts the URL's of the web pages stored in the hidden web pages repository. These extracted URL's are then passed to the AllURL data structure.

2.3.3 AllURL

It contains the URL's as well as the link information of every web page. This information helps in calculating the revisit frequency.

2.3.4 Revisit Frequency Calculator

Incremental Web Crawler main aim is to keep the hidden web page repository fresh. In order to maintain the freshness of page repository incremental web crawler uses Revisit Frequency Calculator. The change of web page is directly proportional to revisit frequency calculator only up to a certain limit. After this limit or threshold value (T) the revisit frequency remain constant.

2.3.5 The Update Module

It computes the inverse of the revisit frequency (τ). If τ of a given URL is greater than T, then that page needs to be re-crawled. These pages are then stored in URL buffer. If τ of a given URL is not greater than T, then that URL will not be re-crawled and will not be stored in URL buffer. Finally the URL buffer will contain the list of all URL's the need to be re-crawled. When this buffer will become full the Update Module will send a signal "Fetch URL" to the Dispatcher module.

2.3.6 Dispatcher

The dispatcher module on receiving the signal "Fetch URL" fetches the URL from the URL buffer. DSHWC then re-download these web pages from these URL's.

Advantages of Incremental Web Crawler

- Incremental Web Crawler keeps the web page repository fresh.
- The non-updated pages are not crawled repeatedly.

Disadvantages of Incremental Web Crawler

- The web pages from both the surface and the hidden web pages are downloaded by Incremental Web Crawler. Therefore mass storage is needed to accommodate these pages in repository.
- Indexing needs to be done.
- Millions of URLs of hidden web pages for even few sites needs to be indexed.

3. HIDDEN WEB SEARCH ENGINES

The different web crawlers are created by the researchers to put the deep web on the surface. In this section we are going to discuss the different hidden web crawlers, their advantages and their disadvantages.

3.1 HiddenSeek

Ntaulas developed a search engine called HiddenSeek[6]. HiddenSeek works on single-attribute database. It also detects the spam websites and ranks the pages also. HiddenSeek is a Web search engine that employs linguistic analysis to improve search relevance.The main components of the HiddenSeek are given below.

3.1.1 Crawler

HiddenSeek has a distributed crawler. The crawler uses the sampling-based policy to maintain a fresh subset of the Web pages with a least overhead. It downloads the pages from the hidden web and then extracts all the links from those pages. Then these URL's are then send to the link database.

3.1.2 Link Database

The link database assigns a unique global ID to every URL identified by the crawler. Along with the URL certain other properties of the URL are also stored for ex: incoming links, incoming outlinks, quality of text etc. This information of the URL is used to rank the web pages later.

3.1.3 Linguistic Processing

The web pages downloaded by crawler are then analyzed linguistically. The linguistic processor uses the natural language processor techniques to improve search relevance.

3.1.4 Inverted Index

Inverted Index stores the ID's of all the web pages for every term. So every record contains the term and the list of ID's that contain that term.

3.1.5 Page Summarizer

This module stores the content of downloaded web pages in a compressed manner.

3.1.6 Answering a Query

HiddenSeek uses the keyword and phrase matching to fetch the result pages. The ANDing of keyword is done. The web pages that contain the user keyword are ranked and sorted. So the important web pages are displayed to user first. For ranking the frequency of keyword in web pages, the position of keyword in web page, the font style of keyword in webpage etc are used to rank the webpage.

3.1.7 User Interfaces

In HiddenSeek a single search box is used. User type the query in the box and the pages indexed earlier by the HiddenSeek, that matches the keyword are displayed to the user.

Advantages of HiddenSeek

- It brings the pages from deep web and serve them to the user.
- It keeps it web page repository fresh by using the sampling-based policy.
- It detects the spam websites also. Their techniques can effectively capture 86% of spam with 91% accuracy.

Disadvantages of HiddenSeek

- It does not work on multiple- attribute databases.
- Millions of queries need to be submitted to get the Hidden Data
- Most websites containing hidden data update very frequently. So crawling has to be done frequently.
- Mass storage will be needed to store web pages from the surface web and hidden web. Please do not revise any of the current designations..

3.2 Hidden Web Search Engine

Dr. Anuradha has created a Hidden Web Search Engine[7]. The hidden web search engine auto fills the web query interfaces, extracts the result records, store them in a repository for later searching. This search engine is able to extract, index and store data sources with multi-attribute interfaces. The main components of Hidden web Crawler are:

3.2.1 Search query Interface Processing & Form Submission

This component is basically employed to work on web query interfaces. It automatically detects the domain specific search interfaces by looking for domain ontology in the source code of the web page. The steps followed in this component are:

3.2.2 Domain Ontology

The domain ontology is created. It is used to create a information architecture in a specific domain. It later provides a knowledge base for classification of search results. To create the domain ontology the attributes are extracted from the downloaded documents. Using the wordnet dictionary the synonyms of the above extracted attributes are identified and are stored in the ontology table. The query interfaces of the desired domain are selected by matching the source code of web pages with the attribute list and their synonyms which are stored in the ontology table.

3.2.3 Unified Query interface

The unified query interface is created by using the information stored in domain ontology. Attribute Preprocessing and attribute matching are done for creating the unified query interface.

3.2.4 Filling Attribute-Value Database

The attribute-value database is filled with the values which later assist in auto form filling. This database contains one table for each attribute and one main table that take cares of the unique identifiers for that attributes. First table in the database contains all attributes under Attribute_name column and id's under Attribute id field.

The second table contains the different attribute values for respective Attribute id under Attribute_value column.

3.2.5 Form Submission

The unified query interface is filled by extracting for each attribute, their field values. The values filled in Unified Query interface are mapped with the original query interfaces and then the original query interfaces are also filled.

Queries are then submitted and the result pages are extracted. The result pages are stored in the web page repository of the search engine.

3.2.6 Hidden web Data Extraction, Integration and Searching

Most of the result pages collected above contains the data in structured relational tables. The proposed hidden web search engine find the information from various hidden web sources and after some processing, present it as a free web search service.The task is conducted by following the steps given below.

3.2.7 Table Area Detection and Extraction

Using the DOM of the HTML pages the table tag are identified. The data from the hidden web is displayed to user in a structured manner using the table. So this component detects the table area, extract the relevant area and discard the other ones. It selects the desired table area then this table area is sent for record area extraction.

3.2.8 Dynamic Rule Generation

The relevant area extracted above is of two types. There are one type of tables where the few details of a product are clubbed in one cell. The information is displayed by product company takes human user into the consideration and not the extraction programs.

The other type of tables have different cell for every detail. From these types of tables it is easy to extract the data. Two different procedures are deployed to extract data. First rule is used for tables like table 2. It extracts data from HTML pages by retrieving all child nodes of each area. New table is created that will consists of first row as the first area where columns are filled by its child node values. The rest of the rows are maintained by rest of the areas and their columns are filled by their respective child node values.

Table1. Web page containing data in table

Description	City	Year	Mileage (km)
Maruti Suzuki WagonR LXI Price :150000 Dealer: Abc Phone: 12345	Faridabad	2000	1000

Table2. Web page containing data in table

Model	Price	Dealer	Phone	City	Year	Mileage (km)
Maruti Suzuki Wagon R LXI	150000	Abc	12345	Faridabad	2000	10000

For each website, one table is created and filled according to the data packed inside its result page.If we have n websites, then we will have n tables respective to each website. For the website that shows data as shown in table1, the data should be collected and semantically labeled so that they can be appropriately organized into main repository which will be used in later searching. For example, ―Maruti Suzuki WagonR Lxi should be inserted into make-model column of main repository and Rs. 150000 should be inserted into price column.

3.2.9 Data Extraction

After extracting rows and columns, separate tables are created for each web site that contains the result records. These tables at the end are merged into the repository. This repository is used to find the result data corresponding to user's query.

Advantages of Hidden web Search Engine

- The results from the hidden web repository are fetched automatically and user can search his data from this repository.
- It works on Multi-valued attributes also.
- It stores the databases from multiple web servers in its local repository and provides a web-service of information sharing.

Disadvantages of Hidden web Search Engine

- Millions of queries need to be fired to fetch data from deep web.
- In order to keep the repository fresh the same set of queries need to be issued at a regular interval.
- Mass storage will be needed to store the data from the large number of hidden websites.

4. HIDDEN WEB WRAPPER GENERATION

There are certain techniques which are ample to download the pages from the hidden web. These downloaded pages from hidden web contains both the relevant data i.e. data from deep web and the irrelevant data i.e. advertisement etc. So there is a need to filter the relevant data from irrelevant data. To accomplish this there are techniques which extract these relevant data from hidden web pages. These techniques are known as wrapper techniques. By extracting the hidden data from multiple sites, the integration of that data can be done which can help in comparative shopping, customizable web information. The integrated information can also be provided as a web service. In this section we are going to discuss the different hidden web wrapper generators, their advantages and their disadvantages.

4.1 Mining data records (MDR)

Mining data records (MDR) [8] proposed by Chen. MDR searches for relevant data by looking for the form and the table tag in the web page. MDR is based on two characteristics of web pages

1) The data areas, area where records from deep web are presented, are in a formatted html tags. The records from deep web in a web page appear continuously.

2) Every web page is structured using the DOM model. DOM presents an HTML document as a tree-structure.

MDR is based on these two characteristics of a web page.

4.1.1 Tree Generator

Using the nested structure of HTML tags, the tree is constructed

4.1.2 Mining Data regions

This phase mines every data region in a page containing similar data records. To get the data region MDR extracts the nodes which have many sibling nodes and all these siblings have one common parent node. MDR terms those nodes as generalized node. These generalized nodes have same parent, have same length and they are adjacent.

4.1.3 Determining Data Regions

This method identifies each data region by finding its generalized nodes. The algorithm basically uses the string comparison results at each parent node to find similar children node combinations to obtain candidate generalized nodes and data regions of the parent node.

4.1.4 Identify Data Records

After all data regions and their generalized nodes are found from a page, MDR identify data records in each region. A generalized node may not be a data record containing a single object. The actual records may be at a lower level, i.e., a generalized node may contain one or more data records. Figure 2 shows a data region that contains two table rows (1 and 2). Row 1 and row 2 have been identified as generalized nodes.

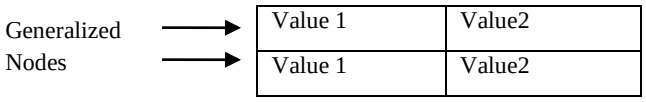

Fig2. Data region containing records

However, they are not individual data records. Each row actually contains two data (objects) records in the two cells.

Advantages of MDR

- It will mine the data records automatically.
- It can be integrated with the hidden web crawlers and search engines, and then it will retrieve the relational data behind the query interfaces.

Disadvantages of MDR

- MDR, not only identifies the relevant data region containing the search result records but also extracts records from all the other sections of the page, e.g., some advertisement records also. The advertisement records are irrelevant.
- Mass storage will be needed to store all the data extracted.

4.2 Layout Based Data Region Finding

The "Layout Based Data Region Finding" is a wrapper technique proposed by Chen. This system [9] has three components. They are:

4.2.1 HTML Tree Constructor

It is designed to translate the HTML file to a Tree, which is the input of Data Region Finder.

4.2.2 Data Region Finder

It takes the Tree as an input, and adopts the LBDRF algorithm to find data region in the list page.

4.2.3 Wrapper Induction

It produces the wrapper rule for data record extraction according to tag path schema.

Advantages of Layout Based Data Region Finding

- It extracts the data from hidden web sources by constructing tree.

Disadvantages of Layout Based Data Region Finding

- This method assumes that large majority of web data records are formed by <table>, <TR> and <TD> tags. Hence, it mines the data records by looking only at these tags. Other tags like , are not considered.

5. COMPARISON OF DIFFERENT HIDDEN WEB DATA EXTRACTION TOOLS

Table3. The comparison of Hidden Web Data Extraction Tools

Type of Tool	Method	Advantages	Disadvantages
Hidden Web Crawlers	DeepBot	The system deals with both client-side scripting code and server side deep web data.	Domain Definitions need to be maintained for issuing meaningful query. Need mass storage to keep hidden web pages
Hidden Web Crawlers	HIWE	Hidden Web crawler will crawl and extract content from hidden databases.	HIWE is unable to identify the relationship between two components of the same form and human assistance is needed to fill the forms Need mass storage to keep hidden web pages
Hidden Web Crawlers	Incremental web Crawler	Adjusting the time period between the two successive revisits of the crawler based on probability of updation of a web page.	The page repository needs to be updated very frequently
Hidden Web Search Engine	HiddenSeek	Uses different query selection policies. Take care of spam websites.	Work on single attribute databases. Need mass storage to keep hidden web pages.
Hidden Web Search Engine	Hidden Web Search Engine	This search engine is works on multi attribute interfaces. The information gathered is compiled and normalized and is presented as a free web search service.	Millions of queries need to be fired to fetch data from deep web. Mass storage will be needed to store the data from the large number of hidden websites.
Wrapper Technique	Mining Data Records	If the web page contains table tags then it can mine the data records automatically.	MDR, not only identifies the relevant data region containing the search result records but also extracts records from all the other sections of the page, e.g., some advertisement records also, which are irrelevant.
Wrapper Technique	Layout Based Data Region Finding	Extracts the data from hidden web sources by constructing tree.	This method assumes that large majority of web data records are formed by <table>, <TR> and <TD> tags. Hence, it mines the data records by looking only at these tags. Other tags like , are not considered.

6. REFERENCES

[1] Chris Sherman and Gary Price Hidden Web. "Uncovering Information Sources Search Engines Can't See". CyberAge book November 2001.

[2] Bergman, Michael K. White Paper. "The Deep Web: Surfacing Hidden Value". Journal of Electronic Publishing Volume 7, Issue 1, August, 2001.

[3] Manuel Álvarez, Juan Raposo, Fidel Cacheda and Alberto Pan. "A Task-specific Approach for Crawling the Deep Web". Engineering Letters, 13:2, EL_13_2_19 (Advance online publication: 4 August 2006).

[4] S.Raghavan and H. Garcia-Molina. "Crawling the hidden web". VLDB, 2001.

[5] Rosy Madaan, Ashutosh Dixit, A.K. Sharma and Komal Kumar Bhatia . "A Framework for Incremental Hidden Web Crawler" International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, 753-758.

[6] Ntoulas Petros Zerfos Junghoo Cho. "Downloading Hidden Web Content". Technical report, UCLA.

[7] Anuradha and A.K Sharma. "Design of Hidden Web Search Engine" International Journal of Computer Applications (0975 – 8887) Volume 30– No.9, September 2011.

[8] Chen Hong-ping; Fang Wei; Yang Zhou; Zhuo Lin; Cui Zhi-Ming; Automatic Data Records Extraction from List Page in Deep Web Sources; 978-0-7695-3699- 6/09 c 2009 IEEE pages 370-373.

[9] Bing Liu, Robert Grossman, and Yanhong Zhai. Mining data records in web pages. In KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 601–606, NewYork.