**Master’s Project Proposal**

**CS700B**

**Creating a Database from Deep Web Sources**

Submitted to the

Department of Computer Science

College of Computing Sciences

New Jersey Institute of Technology

In Partial Fulfillment of

The Requirements for the Degree of

Master’s of Science

By

Jalaj P Asher

jpa7@njit.edu

|  |  |
| --- | --- |
| Proposal Number |  |
| Agree to Advise | **Dr. James Geller** (*Project Advisor*) |
| Date Submitted |  |
| Approved By | (*MS in CS Committee*) |
| Date Approved |  |

**Abstract**

The `Ontology Enabled Web Search` project aims at utilizing the vast information extracted from the Deep Web in assisting users in their search for Web pages. This is done by increasingly returning the number of relevant Web sites. To that end, the project aims at mining the Deep Web for useful information, building domain specific ontology with that information and finally providing the users with a friendly interface for analyzing the search terms entered and getting relevant Web sites to show the contribution of the enriched ontology to Web search effectiveness. Mining the Deep Web for relevant instances based upon a particular ontology is an important process as this information is used in the development and enriching of the ontology and thus making them more usable for indexing of the Deep Web sites.

This project is joint work with Mansi Pedgaonkar and Anushri Mahajan. My area of concentration is creating a database by mining the Deep Web for useful information about famous people for example in the area of healthcare. I will gather a list of people along with some useful information, such as their line of work, age etc. Previously, the mining of the Deep Web for ‘singers’ and ‘basketball players’ was done by mining on a particular website and then gathering relevant information from it [12].Before that the mining of the Deep Web for the `Airports` ontology was done by implementing a parser that gathers information from a Deep Web site by filling out the surface forms depending on the structure of that particular Web page [2].

I will be developing a Deep Web Mining tool trying to avoid non minable Web sites based on the kind of error messages they show. Also will be mining relevant Web sites, get relevant information and build a database depending on the type of ontology of famous personalities. I will also develop a filtering module that will have the information mined as the input and parse relevant information from that by making use of the information stored inside the Deep Web itself. This information is then used for building and enriching the ontology for the `Ontology Enabled Web Search`.

**Table of Contents**

Abstract....................................................................................................................................3

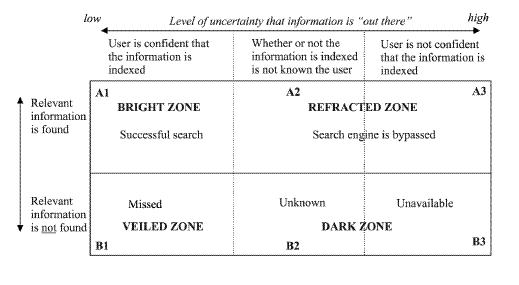
1. Introduction……………………………………………………………………………...5
   1. What is Deep Web………………………………………………………………..…5
   2. Ontologies………………………………………………………………………...…7
   3. Deep Web Mining………………………………………………………………...…9
   4. Role of Ontologies in search for Web pages………………………………………..11
2. Previous Work…………………………………………………………………………...13

2.1 Enriching Ontology for Deep Web Search…………………………………………..13

2.2 Automatic Attribute Extraction from the Deep Web Data Sources…………………14

1. My Work…………………………………………………………………………….…..16
   1. Scenario………………………………………………………………………….…...16
   2. Methodology…………………………………………………………………….…...16
   3. Deliverables………………………………………………………………………….18
   4. Weekly Plan………………………………………………………………………….19
2. **Introduction**
   1. **What is the Deep Web?**

Deep Web refers to Web pages dynamically generated via query interfaces implemented as Web forms or Web services in fact Deep Web is the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web) content that is not part of the [surface Web](http://en.wikipedia.org/wiki/Surface_Web), which is [indexed](http://en.wikipedia.org/wiki/Index_%28search_engine%29) by standard [search engines](http://en.wikipedia.org/wiki/Search_engine). A great deal of information may be caught in the internet, but there is a wealth of information that is deep and therefore missed. Most of the Web's information is buried far down on dynamically generated sites, and standard search engines do not find it. Traditional search engines cannot see or retrieve content in the Deep Web. These Web pages do not exist until they are created dynamically as the result of a specific search. The Web information can be classified according to the relevant Web pages returned on a particular search by a user depending on its indexing by Web crawlers as illustrated in Figure 1 below [12].



**Figure 1:** Distribution of the Web pages Search results based on Relevance [11].

Due to its dynamic nature, existing Web crawlers cannot access the Deep Web. Thus, accessing and maintaining the huge amount of Deep Web information remain challenging research issues. Information in Deep Web sites Information is categorized as being either in textual or structured databases. While a textual database needs input keywords for searching text documents, a structured database requires a user to fill in input fields of a query interface [12].

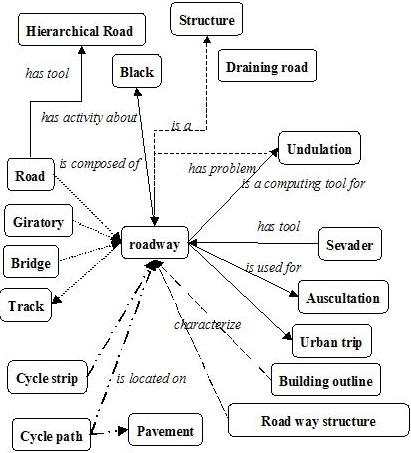
Deep Web content includes information in private databases that are accessible over the Internet but search engines are unable to crawl due to various reasons. For example, some universities, government agencies and other organizations maintain databases of information that were not created for general public access. Other sites may restrict database access to members or subscribers [12].

The Deep Web contained somewhere in the vicinity of 900 billion pages of information. In contrast, Google, the largest search engine, had indexed just 25 billion pages [8].The term, "Deep Web," was coined by BrightPlanet, an Internet search technology company that specializes in searching deep Web content. In their 2001 white paper, 'The Deep Web: Surfacing Hidden Value,' BrightPlanet noted that the Deep Web was growing much more quickly than the Surface Web and that the quality of the content within it was significantly higher than the vast majority of Surface Web content. Although some of the content is not open to the general public, BrightPlanet estimates that 95% of the Deep Web can be accessed through specialized search [9].

**1.2 Ontologies**

Ontology in [computer science](http://en.wikipedia.org/wiki/Computer_science) and [information science](http://en.wikipedia.org/wiki/Information_science) is a formal representation of a set of concepts within a [domain](http://en.wikipedia.org/wiki/Domain_of_discourse) and the relationships between those concepts. It is used to [reason](http://en.wikipedia.org/wiki/Reasoning) about the properties of that domain, and may be used to define the domain. In the words of Thomas Gruber ontology is an explicit specification of a conceptualization. A conceptualization is an abstract, simplified view of the world that we want to represent. If the specification medium is a formal language, the ontology defines a representational foundation [2].

Example: Figure 2 gives a pictorial representation of roadway ontology which consists of various instances and there corresponding relations.



**Figure 2:** An example of roadway ontology [10].

According to Dr. James Geller’s more precise and detailed definition of ontology it is a graph (the data Structure). Every Node of this graph stands for a “concept” which is a unit that one can think about and corresponds to words or short phrases. Typically, a concept corresponds to a noun or noun phrase like house, man, car, New York etc. but that is not an obligation [1].

The nodes of the ontology are connected by different kinds of links. The most important kind of link is called IS-A link. The nodes and IS-A links together form a Rooted Directed Acyclic Graph (Rooted DAG).Rooted means that there is one single "highest node" called the Root. All other nodes are connected by one IS-A link or a chain of several IS-A links to the Root. In our definition, IS-A links point upward. If an IS-A link points from a concept X to a concept Y that means that every real world thing that can be called an X also can be called a Y. In other words, every X IS-A Y. (Some people have IS-A-like links but pointing downwards.) Examples: A car IS-A vehicle. A dog IS-A animal [1].

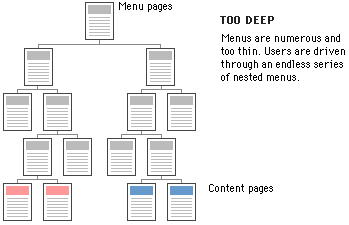
Acyclic means that if you start at one node and move away from it following an IS-A link, you can never return to this node, even if you follow many IS-A links. Most nodes also have other information attached. This information includes attributes, relationships and rules [1].

Ontology represents information in a form that can be used for some forms of reasoning that is at least partially similar to human reasoning. This includes inheritance reasoning, transitivity reasoning and classification. A concept may inherit information from several other concepts. This is called multiple inheritance. Transitivity reasoning corresponds to chaining of IS-A links. Classification means that if we know the attributes of a concept we can decide under which other concepts it belongs in the ontology [12].

**1.3 Deep Web Mining**

Just because a Web search engine can't find something doesn't mean it isn't there. The Deep Web is a vast information repository not always indexed by automated search engines but readily accessible to enlightened individuals. The Shallow Web, also known as the Surface Web or Static Web, is a collection of Web sites indexed by automated search engines [12].

A search engine robots or Web crawler follows URL links, indexes the content and then relays the results back to search engine central for consolidation and user query. Ideally, the process eventually scours the entire Web, subject to vendor time and storage constraints. Most of the time information is stored on Web sites in such a way that the user initially comes in contact with what are called Menu pages. Menus are numerous and too thin i.e. they are having basic information and users are driven through an endless series of nested menus in order to reach important information stored on backend inside Content pages as shown in Figure 2 below [12].



**Figure 3:** Information storage format on a typical website [12].

The crux of the process lies in the indexing. A Web crawler does not report what it can't index.

And we know the search result for a particular Web page in terms of its relevance depends greatly on that. This was a minor issue when the early Web consisted primarily of static generic HTML code, but contemporary Web sites now contain multimedia, scripts and other forms of dynamic content. The Deep Web consists of Web pages that search engines cannot or will not index. The popular term "Invisible Web" which refers to Deep Web is actually a misnomer, because the information is not invisible, it's just not indexed by the Web crawler. The Deep Web is five to 500 times as vast as the Shallow Web, thus making it an immense and extraordinary online resource. The major search engines together index approximately 20% of the Web, and thus missing 80% of the content [12].

Search engines typically do not index the following types of Web sites:

* Proprietary sites
* Sites requiring a registration
* Sites with scripts
* Dynamic sites
* Ephemeral sites
* Sites blocked by local webmasters
* Sites blocked by search engine policy
* Sites with special formats
* Searchable databases

Proprietary sites require a fee. Registration sites require a login or password. A Web crawler can index script code (e.g., Flash, JavaScript), but it can't always ascertain what the script actually does and the Web crawler may get trapped within infinite loops. Dynamic Web sites are created on demand and have no existence prior to the query and limited existence afterward [12].

Webmasters can request that their sites not be indexed (Robot Exclusion Protocol), and some search engines skip sites based on their own inscrutable corporate policies. Not long ago, search engines could not index files in PDF, thus missing an enormous quantity of vendor white papers and technical reports, not to mention government documents. Special formats become less of an issue as index engines become smarter. The most valuable Deep Web resources are searchable databases. There are thousands of high-quality, authoritative online specialty databases. These resources are extremely useful for a focused search [12].

**1.4 Role of Ontologies in search for Web pages**

Recently, there has been a growing interest in Web searches that are intended to locate information that exists in the backend data bases of Web services. Web sites in E-commerce domains such as airfares, automobiles, books, car rentals, hotels, jobs, movies and music records usually store huge amounts of information, which is of interest to many users, in their backend databases [12].

Information in E-commerce backend databases is usually not “visible” to general search engines. The information in backend databases is often called Deep Web data. Finding the relevant E-commerce sites and accessing, retrieving and indexing the huge amounts of Deep Web data raises challenging research issues [12].

Ontology could play an important role in assisting users in their search for Web pages. Domain ontology can be constructed that support users in their Web search efforts and that increase the number of relevant Web pages that are returned. To achieve this goal the Deep Web information, which consists of dynamically generated Web pages, which cannot be indexed by the existing automated Web crawlers, is combined with ontology.[12]

The process of building ontology consists of several steps as shown in Figure 4. Firstly the possible search terms, called attributes of Deep Web data sources are automatically extracted from a static collection of Deep Web sites. Secondly, separate domain ontology is built for each domain, using the extracted search terms. Thirdly, by probing a few Deep Web sites domain terms from the backend databases are extracted. Next, the domain ontology is extended to include these Deep Web terms as instances. Finally, the domain ontology is extended with relationships between instances [3].



**Figure 4:** The Flow for an Ontology-Supported Deep Web Search [3].

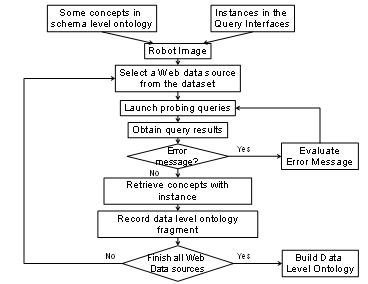
**2. Previous Work**

This is part of an ongoing project. The previous work on this project included addressing the problems of extracting instances from the Deep Web, enriching a domain specific ontology with those instances, and using this ontology to improve Web search. The first domain to be mined, for ontology creation, was ‘Airports’. The list contained airports of North America only. The ontology was used to provide better search terms for users to have more relevant search results for users and it was compared with the search terms provided by the user to search engines like Google, Yahoo etc. In the second continuation of the project the websites were searched manually to get relevant websites that could give info on singers and basketball players. A website was then chosen on which Deep search techniques are used to get information on singers and basketball players. The information then derived was saved in database. This information is then compared with sample information to determine the accuracy of it before being considered for making ontology. In the current phase of this project the domain has been extended to searching out information about healthcare specialists, North American Companies etc.

**2.1 Enriching Ontology for Deep Web Search**

Enrichment of ontology is a process that extends it by adding concepts, instances and new relations between concepts. In the related paper method for extracting instances from the DW is based on developing “robots” (agents) that sent many queries to the same DW site to extract as many data values as possible. When a robot encounters an input field it may enter random values or leave the field empty and then submit the page to elicit an informative response [3].

Figure 5 shows the workflow for extending the ontology with instances. The concept discovery of the robot is guided by a human in its initial stage. Initial pairs of a concept and its corresponding instances are defined, which we call a *robot image*. The robot submits input values into the query interface. If the input values are not suitable for the form, most Web sites display error messages. The analysis of the error messages often gives useful clues to the robot to guess suitable input values and launch better probing queries. Thus, the queried Web sources may provide information about concepts, instances and semantic relationships, which is recorded in the ontology [3].



**Figure 5:** A Flow for generating data level ontology fragments [3].

**2.2 Automatic Attribute Extraction from the Deep Web Data Sources.**

Understanding the attributes and contents of Deep Web data sources is also important in order to locate the most relevant Deep Web data sources for a user, since these sources use different attributes to access contents. The related paper presents a novel approach to automatically extract attributes from query interface in order to address the current limitations in accessing Deep Web data sources. It introduces the Semantic Deep Web for utilizing ontology to determine attributes to access the Deep Web [4].

In the general sense, an attribute of a Web data source is any item of information that describes this source. The more specific meaning of “attribute” is derived from the HTML/XML syntax. A tag of HTML consists of a mandatory name between angular brackets, which may be followed by optional attribute/value pairs. As an example, the Web page in Figure 6 is generated by the HTML code in Figure 7, which contains several attribute/value pairs. Thus the tag <SELECT> contains the attribute “size” with the value 2. The attribute and the value of such a pair are separated by an equal sign (=) [4].



**Figure 6:** An Example of a simple Form [4].

<FORM action= “...” method= “…”>

<P><LABEL for= “departure\_city”>Departure<BR>

</LABEL>

<SELECT size= “2” name= “depart\_city”>

<OPTION selected value= “city1”>Newark</OPTION>

<OPTION>Arlington</OPTION></SELECT><P>

Where is your departure city? <BR>

<INPUT type= “text” id= “origin”><P> Search by:<BR>

<INPUT type= “radio” name=“searchBy” value=“fare”> fare<BR>

<INPUT type=“radio” name=“searchBy” value= “schedule”>

schedule<P>

<INPUT type= “submit” value= “Go”> </P> </FORM>

**Figure 7:** HTML corresponding to Figure 6 [4].

1. **My Work**

**3.1 Scenario**

This target of this project is to mine instances from the Deep Web, and then enrich a domain specific ontology with those instances, and finally using this ontology to improve Web search for users. My work in the project is to develop a Deep Web Mining tool for gathering the relevant information. Classify Web sites based on the error messages that we get on querying them like forbidden access, page not found etc. Determine the Web sites we can mine on and remove those that we can’t. Build a database depending on the type of ontology to be built which right now refers to `Singers` and ‘Basketball players’ can be further extended to `healthcare specialists`, `National Football League` etc.

The second part of the project involves a filtering module that will take the information generated and filter out any duplicate or incorrect information again by making use of the information stored inside the Deep Web itself. This information is then used for building and enriching the ontology for the `Ontology Enabled Web Search’.

**3.2 Methodology**

I will develop a program for extracting instances from the Deep Web by sending many queries to the several Web sites to extract as many data values as possible. The first step would be to find minable Web sites that have information regarding ‘relevant topic’. This step includes classifying the errors encountered from different Web sites. Group them as non minable or change input parameters for error messages ‘wrong input error’. The program then submits input values into the query interface by analyzing the structure of that particular Web page and extracting as much details of the topic, as it can, here healthcare specialists. This information will then be entered in a database. The work flow for this process is also depicted in Figure 8.

The limitation of this extraction is that even though it can get the details on the topic we need to verify how authentic that information is. Also when we say healthcare specialists it could relate that even to companies or engineers that work in the health care sector but we have to make sure it is relates only to doctors, specifically.

For this the information extracted from the Deep Web needs to be filtered and that will be the part of my other module. It will take the information generated by the program from the database and then select some Deep Web site that has information about all the healthcare specialists, their name, sex, style, locations etc. This information will not only help us to filter out relevant details from unnecessary information collected from the Web sites but will also help us gather important instances for the domain ontology. This will be done by parsing the Web pages and then analyzing their structure i.e. the attribute /value pairs of the HTML code as it defines the only way for accessing the form information.

Error classification

**Deep Web**

Remove non -minable Web sites

Module to mine Deep Web

Healthcare database

Database

Filtering and Information Enrichment Module

**Figure 8:** Flow of Data Extraction and Filtering Process

* 1. **Deliverables**

1. **Error classification**

* Manually check for errors on Web site.
* Classify those errors as information missing, invalid input, information not available, page missing or forbidden access to the system.
* Incorporate this information in the Deep Web Website Extractor application.

**2. Deep Web Website Extractor (**an application for extracting details of healthcare           specialists.**)**

**Input:** Deep Web site containing names of famous personalities.

**Process:**

* Parse the website for finding all the attributes/values pairs.
* Analyzing the structure of a particular Web page.
* Submit queries according to the information to be extracted.
* Analyze errors, if generated during the query based on the classification of websites as provided by the previous deliverable. Try to group Web sites according to errors and modify queries accordingly. Discard those Web sites that cannot be mined.
* Extract names.
* Connect to Oracle database.
* Enter the gathered names into the database.

**Output:** Database containing names of healthcare specialists.

**3. Database Filtering and Enriching Module.**

**Input:** Names of the healthcare specialists in database.

**Process:**

* Connects to the database and retrieves the names of healthcare specialists gathered.
* Connect to a suitable Deep Web site for filtering out the names for any incorrect information collected by again parsing that particular Web site.
* If the name is valid then gather any additional information like age, sex, location, specialization etc.
* Store the information extracted into the database along with the person’s name.

**Output:** Required list of healthcare specialists’ and additional information.

**3.4 Weekly Plan**

|  |  |
| --- | --- |
| Week 1 | Getting Started: Studying work done previously, the limitations and problems faced previously. |
| Week 2 | Researching various Deep Web sites for related information about healthcare specialists and find the errors that Web sites generate and try to classify those errors, understand the general structure of sites and Web technologies that can be helpful. |
| Week 3 | Continued. |
| Week 4 | Writing the code for the **Deep Web Sites Extractor** for extracting the names of healthcare specialists by mining the Deep Web. |
| Week 5 | Continued. |
| Week 6 | Adding the information collected to the database and optimizing the code. |
| Week 7 | Writing the code for the Database Filtering and Enriching Module. |
| Week 8 | Continued. |
| Week 9 | Connecting the Database Filtering and Enriching Modulewith the healthcare specialists’ database and gathering additional information. |
| Week 10 | Extending the Domain to National Football League Players. |
| Week 11 | Continued. |
| Week 12 | Testing the Deep Web sites Extractor and Database Filtering and Enriching Module. |
| Week 13 | Continued. |
| Week 14 | Writing Report for the work done. |

**References**

1. James Geller. Definition of Ontology,

<http://web.njit.edu/~geller/what_is_an_ontology.html>, February 16, 2009.

1. **Thomas Gruber (2008),** [**Ontology**](http://tomgruber.org/writing/ontology-in-encyclopedia.htm)**. Entry in the**Encyclopedia of Database Systems***,* Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag. February 16, 2009.**
2. Yoo Jung An, [Soon Ae Chun](http://www.informatik.uni-trier.de/%7Eley/db/indices/a-tree/c/Chun:Soon_Ae.html), [Kuo-chuan Huang](http://www.informatik.uni-trier.de/%7Eley/db/indices/a-tree/h/Huang:Kuo=chuan.html), [James Geller](http://www.informatik.uni-trier.de/%7Eley/db/indices/a-tree/g/Geller:James.html): Enriching Ontology for Deep Web Search. [DEXA 2008](http://www.informatik.uni-trier.de/%7Eley/db/conf/dexa/dexa2008.html#AnCHG08): 73-80, January 24, 2009.
3. Yoo Jung An, James Geller, Yi-Ta Wu, Soon Ae Chun: Semantic Deep Web-Automatic Attribute Extraction from the Deep Web Data Sources, March 12, 2009.
4. Yoo Jung An, Soon Ae Chun, Kuo-chuan Huang, James Geller: Assessment for Ontology Supported Deep Web Search, March 14, 2009. March 7, 2009.
5. Yagnesh Bhat, Building Ontologies for Ontology Enabled Web Search, NJIT, CS Department, MS Proposal Spring 2009.
6. Tian Tian,Enriching Multi-Domain Ontology for Web Search, PhD Project Proposal, NJIT, CS Department, in preparation of Ph.D.
7. <http://en.wikipedia.org/wiki/Google_search>, February 21, 2009.
8. <http://brightplanet.com>, February 21, 2009.
9. <http://www.cybergeo.eu/docannexe/image/8322/img-1.jpg>, March 12, 2009.
10. <http://www.wowter.nl/blog/uploaded_images/ford-795385.gif>, March 15, 2009.
11. Ashish Rajbar, Deep Web Mining and Enriching the Ontology Enabled Web search, NJIT, CS Department, MS Proposal Spring 2009.