# EXTRACTION OF LABELS FROM SEARCH INTERFACES FOR DOMAIN SPECIFIC HIWE

Shikha Tomar[1], Varun Kaushik[2]
[1]*Department of Computer Engineering, TITM, Meerut, UP, India*
[2] *Department of Computer Engineering, MIET, Meerut, UP, India*

**Abstract:** Over the years, the surface web Publicly Indexed Web) has grown to billions of HTML pages. Simultaneously, the web has been rapidly "deepened" by the massive web data-sources and a far more significant amount of information is hidden in the deep web, behind query forms of web data-sources like amazon.com. These data-sources are also referred as hidden or invisible web. These hidden sources allow the users to access the underlying information by querying through their query interfaces. Where in data-sources contain the attributes that tend to describe the information accessible through them. For example, the query interface of a source like amazon.com contains attributes such as author, title, ISBN etc. These query interfaces acts as the entry points to the hidden or invisible web and therefore became potential candidates to be identified to extract hidden data. This paper presents a mechanism to extract pages and forms and identifying search forms and extracts out their labels which could be further used by the hidden web crawler.

## 1. INTRODUCTION

A search engine is an information retrieval system designed to help to minimize the time required to find information over the vast web hyperlinked document. When a user enters a query into a search engine, the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. The crawler retrieves web pages commonly for use by a search engine. Web crawling is used by Search engines to provide up-to-date data to the users. It traverses the web by downloading the documents and following embedded links from page to page. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them. Large portion of the web are 'hidden' behind the search forms in searchable databases. These are known as 'hidden web' and also referred as 'Deep web'. The fact that search engine only search a very small portion of the web make the Invisible Web a very charming resource. There's a lot more information out there that can be imagined that's because terabytes of information are immersed in databases and other research resources. Search engine that use traditional crawlers never find this information, so required to build a hidden web crawler which can crawl and extract content from hidden databases.

### 1.1 HIDDEN WEB

The Hidden Web also called Deep net, the Invisible Web, or the Deep Web. It refers to World Wide Web content that is not part of the surface Web, which is indexed by search engines.

### 1.2 HIDDEN WEB DATABASE MODEL

There exists a variety of Hidden Web sources that provide information on a multitude of topics. Depending on the type of information, we may categorize a Hidden-Web site either as a textual database or a structured database. A textual database is a site that mainly contains plain-text documents. Since plain-text documents do not usually have well-defined structure, most textual databases provide a simple search interface where users type a list of keywords in a single search box (Fig. 1.2a). In contrast, a structured database often contains multi-attribute relational data (e.g., a book on the Amazon Web site may have the fields title='Harry Potter', author='J.K. Rowling' and ISBN='0590353403') and supports multi-attribute search interfaces (Fig. 1.2b).

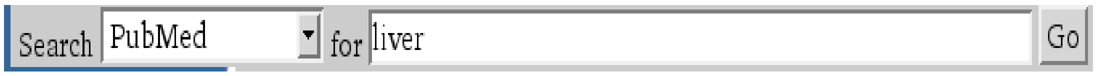


Fig. 1.2a A single-attribute search interface

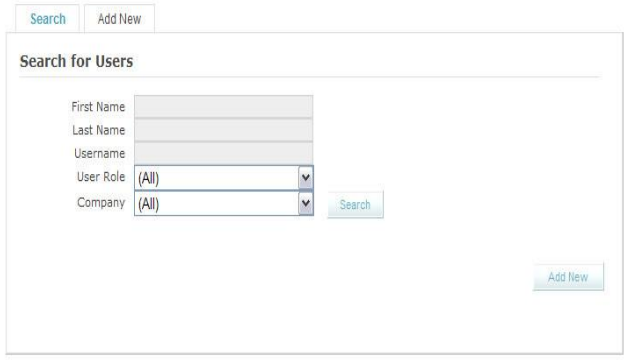
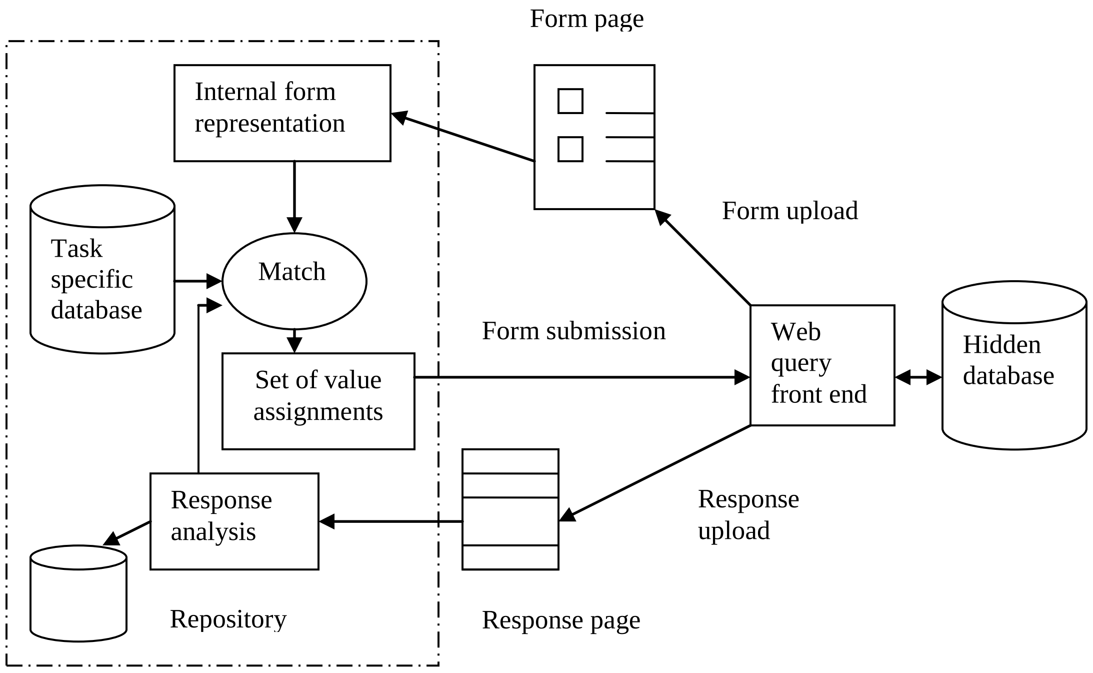


Fig. 1.2b: A multi-attribute search interface

**1.3. EXTRACTION OF INFORMATION FROM HIDDEN WEB**

Large portions of the Web are 'hidden' behind search forms, in searchable structured and unstructured databases (called the hidden Web or deep Web). Pages in the hidden Web are dynamically generated in response to queries submitted via the search forms. The hidden Web crawler can crawl and extract content from these hidden databases. Such a crawler will enable indexing, analysis, and mining of hidden Web content.



. The model of a hidden Web crawler consists of the four components described here. *Form page* is used to denote the page containing a search form, and *response page*, to denote the page received in response to a form submission.

## 2. IMPLEMENTATION OF EXTRACTION OF LABELS FROM SEARCH INTERFACES FOR DOMAIN-SPECIFIC HIWE.

In the proposed methodology a crawler traverse the web by downloading the documents such as files and images and save them in a repository by traversing all hyperlinked documents. Then extract search interface forms from these HTML pages or files, after that extract all the labels from these interface an store them in a file. The main motive of this methodology is to extract labels from the search interface. There are some modules, which are used in this methodology. The brief explanation of all these modules is given in next section.

## 2.1 MODULES USED IN THE PROPOSED METHODOLOGY

In this section, describing the following modules in brief which are used in this architecture. They are-
Download Site- This module is used to download the websites from seed URLs. Seed URLs is an array having some URLs. Download site one by one takes the URLs from this array and download these websites for the use of next module.
Downloader- This is the main module of this methodology, it traverse the website which are downloaded by download site. This module also traverses the hyperlinked documents and store URLs of these hyperlinked in a vector. This vector is named as vector of URLs. It downloads the 'index.html' file of the website first and then all hyperlinked URLs. And store all these URLs in the vector of URLs. It downloads the files and images and stores them in a page repository.
There are two sub modules of downloader-
URLList- This process is used to manage the vector of URLs. It reduces the duplicity. It removes those URLs from the list which are already exists in the vector and also check for their existence.
ExtendedURLList- This is used for the formatting of URLs, which are stored in vector of URLs.
PageFetch- Images and files which are stored in page repository are traverses by this module one by one. It checks weather these files are html file or not If not then discard those files, if they are then forward it to next module. It discards the images and select only html files for the next process.
FormInfoFetch – This module checks those HTML files which are checked by page Fetch. This module checks that these html files contain a form tag or not? If it contains a <form> tag then store that page in a repository named form repository, if not then discard that page. It will save only form in the directory from the html file else part will be removed like images, ads etc.
SearchPageFetch – This module checks for the search interface. It reads the page and check for the search keywords. If the interface have keywords like search, go, find etc. then save these pages in a search repository else discard the pages.
Label Fetch- This module fetches the file from the search page repository and extracts out the labels from the file. This is done by first finding the type attribute of "<input " tag, if the type attribute of "<input" tag is text or checkbox or radiobutton, it means the text before this tag represents a label.

**A PROPOSED DATA FLOW DIAGRAM FOR EXTRACTION OF LABELS FROM SEARCH INTERFACES FOR DOMAIN-SPECIFIC HIWE.**

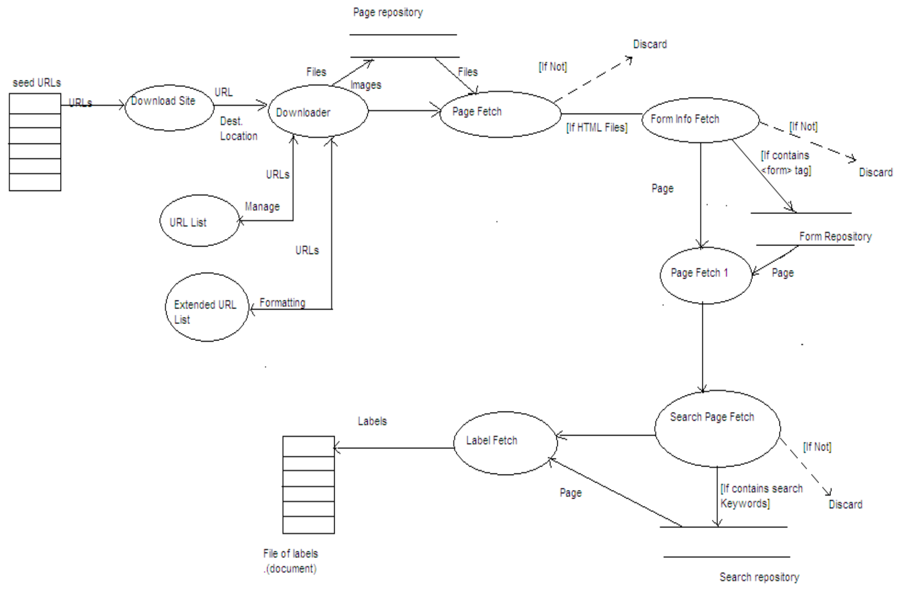


Fig 2.1 Data Flow Diagram for Proposed Methodology

## 3. Conclusion and future scope

As a large amount of web contents residing on hidden web are generated dynamically from databases and other data sources hidden from the user. These web contents are generated only when queries are asked via a search interface, extracting the search interfaces is a critical problem in many application domains, such as: semantic web, data warehouses, e-commerce etc. Many different solutions have been proposed so far. In this paper, a promising approach for extraction of search interfaces has been implemented. It reads the caption of the button and if the caption of the button contains the text such as "go, search, find" etc. than such form is considered as a search interface. The results obtained for specific domain (Book Domain) is highly accurate as the Precision varies from 81% to 100%, Recall varies from 50% to 60% and F-measure varies from 62% to 72%. To the best of our knowledge, this is the first piece of work which makes such a guarantee for search interface extractor. Implementation has been performed on Book domain and the results are found highly accurate.

Future research includes improvement in the technique of parsing the forms by reducing response time. Some of the future works are as follows –

1. After correctly identifying the labels of the search interfaces , tried to find out which component is representing by this label so, that the form is automatically fill out and submitted to fetch the hidden data.
2. We can apply distributed processing to this proposed system by this, proposed system work parallely to more than one seed urls at a time.

## Refernces

[1]. A. K. Sharma, Komal Kumar Bhatia: "Merging query interfaces in domain specific hidden web databases"
[2]. A.K.Sharma and Komal Kumar Bhatia, A Framework for Domain-Specific Interface Mapper (DSIM)
[3]. A. K. Sharma, Komal Kumar Bhatia: "Automated Discovery of Task Oriented Search Interfaces through Augmented Hypertext Documents" accepted at First International Conference on Web Engineering & Application (ICWA2006).
[4]. Luciano Barbosa, Juliana Freire, "Combining Classifiers to Identify Online Databases", WWW2007/track
[5]. T. Mitchell. Machine Learning. McGraw Hill, 1997
[6]. L. Barbosa and J. Freire. "Searching for Hidden-Web Databases". In Proceedings of WebDB, pages 1–6, 2005
[7]. http://en.wikipedia.org/w/index.php?title=Deep_web&redirect=no
[8]. A. K. Sharma, J. P. Gupta, "An Architecture of Electronic Commerce on the Internet", accepted for the publication in the fourth coming issue of Journal of Continuing Engineering Education, Roorkee, Jan 2003
[9]. Dinesh Sharma, A.K. Sharma, Komal Kumar Bhatia, "Web crawlers: a review", NCTC-2007
[10]. Dinesh Sharma, A.K. Sharma, Komal Kumar Bhatia," Search engines: a comparative review",NGCIS-2007
[11]. http://www.cs.utah.edu/~juliana/pub/webdb 2005. pdf
[12]. http://en.wikipedia.org/wiki/Deep_web
[13]. www.invisibleweb.com
[14]. Alexandros Ntoulas Petros Zerfos Junghoo Cho, "Downloading Hidden Web Content", UCLA
[15]. A. Bergholz and B. Chidlovskii. Crawling for Domain-Specific Hidden Web Resources. In Proceedings of WISE, pages 125–133, 2003
[16]. http://en.wikipedia.org/wiki/parsing