Content Discovery from Deep Web using Large Scale Data Analytics Paradigm

By

Rao Muhammad Umer

Thesis submitted to the Faculty of Computer and Information Sciences (DCIS) in partial fulfillment of requirements for the Degree of MS Computer Sciences.

Department of Computer and Information Sciences
Pakistan Institute of Engineering & Applied Sciences,
Nilore, Islamabad
October, 2016

In the Name of ALLAH, Most Gracious,

Most Merciful

Department of Computer and Information Sciences,

Pakistan Institute of Engineering and Applied Sciences (PIEAS)

Nilore, Islamabad 45650, Pakistan

Declaration of Originality

I hereby declare that the work contained in this thesis and the intellectual content of this thesis are the product of my own work. This thesis has not been previously published in any form nor does it contain any verbatim of the published resources which could be treated as infringement of the international copyright law.

I also declare that I do understand the terms 'copyright' and 'plagiarism', and that in case of any copyright violation or plagiarism found in this work, I will be held fully responsible of the consequences of any such violation.

Signature: ______________________________

Name: Rao Muhammad Umer

Date: ______________________________

Certificate of Approval

This is to certify that the work contained in this thesis entitled

"Content Discovery from Deep Web using Large Scale Data Analytics Paradigm"

was carried out by

Rao Muhammad Umer

Under our supervision and that in our opinion, it is fully adequate, in scope and quality, for the degree of MS Computer Science from Pakistan Institute of Engineering and Applied Sciences (PIEAS).

Approved By:

Signature: ________________________

Supervisor: ***Dr. Muhammad Abid Mughal***

Signature: ________________________

Co-Supervisor: ***Dr. Fayyaz-ul-Amir Afsar Minhas***

Verified By:

Signature: ________________________

Head, DCIS

Stamp:

Dedicated

To my Parents for their love and affection, who
prays all the time for me and what am I today,
is due to their encouragement and support.

Acknowledgement

Gratitude and endless thanks to Allah Almighty, the Lord of the World, who bestowed mankind, the light of knowledge through laurels of perception, learning and reasoning, in the way of searching, inquiring and finding the ultimate truth. To whom we serve, and to whom we pray for help.

I feel my privilege and honor to express my sincere gratitude to my supervisor,

Dr. Muhammad Abid Mughal, for all their kind help, guidance, suggestions and support through this project.

I would like to express my most sincere gratitude and thanks to my co-supervisor,

Dr. Fayyaz-ul-Amir Afsar Minhas, whose continuous guidance made me progress through my project work smoothly and his support was always there whenever I stuck somewhere in finding the relevant technical help.

I would also like to thank Department of Computer and Information Sciences (DCIS), Pakistan Institute of Engineering and Applied Sciences (PIEAS), and Pattern Recognition Lab for providing very conducive educational environment in completion of my thesis work.

Finally, I would like to thank all those, whose names have not been mentioned here, but they have helped me in one or the other way in completion of my thesis work.

Rao Muhammad Umer,
PIEAS, Nilore, Islamabad,
October, 2016.

Table of Contents

List of Figures

List of Tables

Nomenclature

DWX	Deep Web Extractor
SVM	Support Vector Machine
CRF	Conditional Random Field
WWW	World Wide Web
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
DW	Deep Web
SGD	Stochastic Gradient Descent
LR	Logistic Regression

Abstract

Exponential growth of Web imposes a major challenge for retrieving and searching data across the Web for many information retrieval and data mining (i.e. text mining) tasks. Search engines such as Google, Yahoo, Bing, etc., rely on link crawling strategies to index the static web pages (aka Surface Web). However, lots of other massive and quality portion of web is hidden from these engines, which is known as the Deep Web. The immense and high quality data form the Deep Web is also useful for building knowledge based databases. Discovering and extracting the vast content from the deep web is a major research challenge for knowledge discovery and information retrieval community. This thesis explores the factors preventing typical search engines from indexing Deep Web contents & provides the solution in such a way that Deep Web contents can be extracted and exposed to web searchers. The proposed DWX system is a cloud based web application for Crawling & Data Discovery from Deep Web. The proposed DWX system allows a web user to set a specific query (e.g. keywords relating to specific topic), and to retrieve the significant information in an effective way using machine learning techniques.

Chapter 1: Introduction

The continuous growth of Web imposes a major challenge for retrieving and searching data across the Web for many information retrieval and data mining tasks. Since web users usually depend upon search engines like Google, Yahoo, Bing, etc., but these search engines merely rely on link crawling the Surface Web and other immense and useful part of web is hidden from these engines, which is known as the Deep Web. Discovering and extracting this vast deep web content is a major research challenge for information retrieval community.

1.1 The Deep Web

The **Deep Web [1]**, **Invisible Web [2]**, or **Hidden Web [3]** is the part of World Wide Web (WWW), whose web contents cannot be indexed by standard search engines for some reason. The deep web is opposite to the surface web.

The Surface Web or Visible Web consists of web pages that are indexed by standard search engines, e.g. Google, Yahoo or Bing. According to recent estimate, there are at least 4 billion webpages indexed by standard search engines. Besides that, there is a much larger portion of the World Wide Web (WWW) that is hidden below the Surface Web, which cannot be indexed by standard search engines. This part of the web is called Deep Web.

Most of the web's information is hidden behind dynamically generated websites, such as web form interfaces. Traditional search engines work on link crawling strategy, so they build their indices by crawling the static webpages for finding new links on web. To crawl pages on a website, pages must be static and linked to other pages. Traditional search engines cannot retrieve web contents form deep Web resources, because those pages do not exist until they are built dynamically as the result of a specific query search by a user. The Deep Web sources store their contents in searchable databases that only produce results dynamically in response to a direct query request.

Now, we take the real use cases to understand the deep web comprehensibly. Let's take an example of “hdmoviespoint.com” website; we can download a particular movie by clicking the download button, which refers to another webpage that contained the download link of a specific movie. In this case, this download link page is only accessible by first going through "hdmoviespoint.com" website. However, google search result for specific movie cannot include this download link page. Therefore, the contents of this page refer to deep web content.

1.2 Objectives of the Project

This project is aimed at the development of Deep Web contents extractor system for web searchers to extract the quality of information from the deep web resources. It is our goal to employ a fully automated approach for extracting and searching the Deep Web resources.

The following are main objectives of this project:

- To discover and extract the deep web's content of quality for web searchers.
- To discover automated means for identifying searchable web form interfaces and directing queries to them to dig out information.
- To build domain specific data repositories (e.g. real estate, newspapers, health, etc.) for purposeful analysis and uncovering hidden knowledge
- To handle the complex queries, like queries containing different range values, not entertained by traditional search engines.

1.3 DWX System Architecture

The proposed architecture of system “DWX” comprises of the following major components as shown in Figure 1-2:

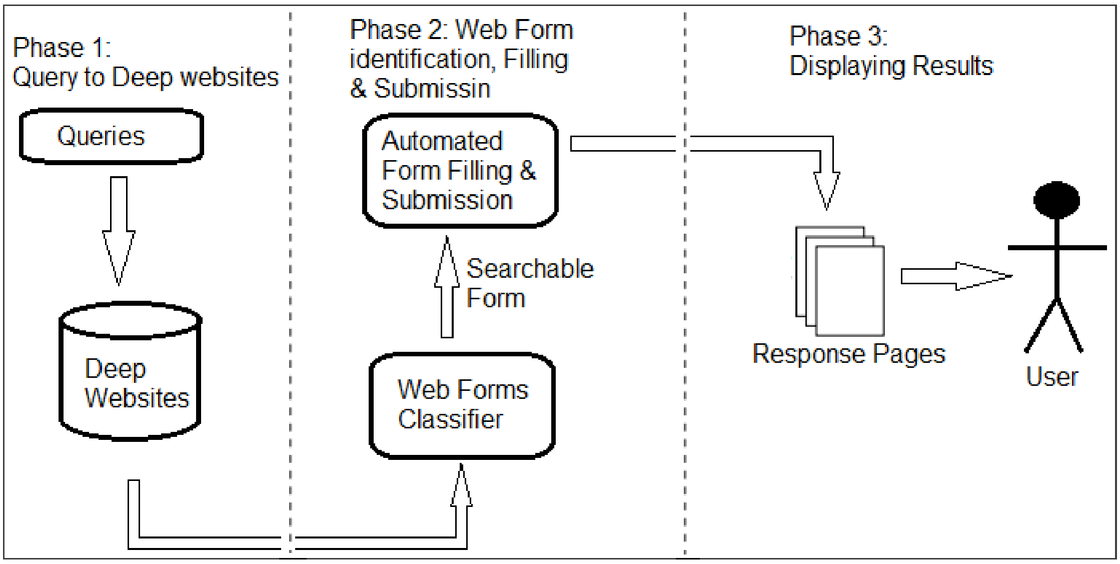

Phase 1:
Query to Deep websites
Queries
Deep Websites
Phase 2: Web Form identification, Filling & Submissin
Automated Form Filling & Submission
Searchable Form
Web Forms Classifier
Phase 3: Displaying Results
Response Pages
User

Figure 1-1: DWX System Architecture

Below a description of each component is presented:

1.3.1 User Interface

It presents a high level interface to the user for passing queries. This interface passes that query to the corresponding deep web resources.

1.3.2 Form Classifier

It identifies the web form in a website and check whether it is a searchable form or not. If the form is a searchable, it will identify the fields in that form.

1.3.3 Automated Form Filling & Submission

The identified fields in the searchable form are automatically filled with user-given query and submitted to corresponding website.

1.3.4 Displaying Results

The returned response by submitting the web form is displayed to user that is searching the required information.

1.4 Organization of Thesis

This document serves as a detailed description of the project and presents in detail, different steps involved in the development of the DWX system. Chapter-1 presents introduction to the project. Chapter-2 provides introduction to related work for Deep

Web extraction. Chapter-3 provides the web forms classification using machine learning techniques. Chapter-4 describes DWX application architecture and its implementation. Chapter-5 describes large scale data extraction using distributed computing framework. Chapter-6 presents the conclusions and future work. Each chapter describes the objectives and importance of the task it addresses and describes in detail the implemented schemes.

Chapter 2: Deep Web Data Extraction

Web data extraction is an important field that heavily depends on techniques and algorithms developed in the field of Information Extraction and Retrieval. Emilio Ferrara el al. [4] have discuss web data extraction applications at the Enterprise level and the Social Web level and also discuss potential purposeful analysis for predicting human behavior on extracted data form web. Similarly, Steve Pederson [5] discussed the usefulness of the Deep Web extracted data. Deep Web extracted data are also useful for many text mining tasks.

2.1 Size of World Wide Web (WWW)

Antal van den Bosch et al. [6] gave useful statistics about the size of indexed web pages of WWW form standard search engines (i.e. Google, Bing). They recently estimated that at least **4.72 billion** webpages are being indexed by typical search engines. The estimated minimal size of the indexed World Wide Web depends upon the estimations of the numbers of webpages indexed by Google, Bing and Yahoo search results. Besides these, there are still billions of webpages not indexed by search engines and reside on the Deep Web.

2.2 Deep Web Resources

Michael k. Bergman [7] estimated the Deep Web size and its relevancy for web searchers. He also presented some useful statistics about the Deep Web size.

Here are the following important findings about the size and relevancy of the deep web:

- Public information is currently 400 to 500 times larger on the deep web than that of World Wide Web.
- Search engines can access only 0.03% of the World Wide Web. The remaining is known as the deep web.
- The deep web consists of 7,500 terabytes of information as compared to 19 terabytes of information in the surface web.
- The deep web comprises of approximately 550 billion individual documents as compared to the one billion documents of the surface web.

- There are more than 200,000 deep web sites presently reside on WWW.
- The deep websites receive 50% greater monthly traffic than surface websites. However, the typical deep web site is not well known to public internet searching.
- The deep web is the largest growing category of new information on the internet.
- Deep websites tend to be narrower with deeper contents than traditional surface websites.
- Total quality content of the deep web is 1,000 to 2,000 times larger than that of the surface web.
- Deep Web contents are highly relevant to every information need, market, and domain specific information.
- There are more than half of the deep web content reside in domain specific databases.

There are many reasons [8] which prevent webpages from being indexed by traditional search engines, some important ones are listed here:

2.2.1 Contextual Web

Contextual webpages are varying and depend on the webpages is varying contexts e.g. ranges of client IP addresses or previous navigation sequence. Search engines cannot index contextual webpages.

2.2.2 Dynamic Content

Search engines cannot index web pages generated at run-time known as dynamic web pages. Dynamic webpages, which are returned in response to a submitted query or retrieve only through a web form, especially hard to navigate if it has open domain input elements e.g. text fields.

2.2.3 Limited Access Content

The websites, that limit access to their pages in a technical way by using the Robots Exclusion Standard or CAPTCHAs, cannot be indexed.

2.2.4 Non-HTML/text Content

Textual contents, which are encoded in multimedia such as image or video files or specific file formats, are not handled by search engines.

2.2.5 Private Web

Websites, that require registration and login information, are not indexed by search engine.

2.2.6 Scripted Content

The webpages, that are only retrieved by links created by JavaScript as well as web content dynamically downloaded from web servers through Flash or Ajax solutions, are not handled by traditional search engines.

2.2.7 Unlinked Content

The webpages, which are not linked by other webpages, can prevent web crawling programs to access their contents. These contents are referred to webpages without backlinks aka inbound links. Search engines do not always detect all backlinks from searched web pages.

2.2.8 Web Archives

Web archival services [9], in which users can see archived versions of web pages across time, are not indexed by search engines such as Google.

2.2.9 Searchable Databases

Searchable databases, which are behind the web form interfaces, are the most valuable resource of deep web.

In this project, the major focus is on the **“dynamic contents”** and **“searchable databases”** for extracting the deep web contents.

2.3 Related Work for Deep Web Data Extraction

James Caverlee et al. [10] developed the “THOR” system for discovering and extracting QA-Pagelets from the Deep Web resources. This system discovers and extracts the deep web on the basis of structure and content similarity of QA-Pagelets.

RoadRunner [11] proposed system automatically creates wrappers for extracting data from web pages and compares HTML pages. It generates a wrapper based on pages resemblances and dissimilarities. RoadRunner Algorithm compares pages produced by the same query form and builds a regular expression based on the dissimilarities between the pages. Similarly, Arasu et al. [12] developed a data extraction method that models template-generated pages and infers the unknown template used to generate the pages. It also mines the values encoded in the pages.

Panagiotis G. Ipeirotis et al. [13] proposed an algorithm to originate content summaries from “uncooperative” databases by using focused query searching that extracts documents which are demonstrative of the topic exposure of the databases.

Luciano Barbosa and Juliana Freire [14] proposed a crawler to automatically locate hidden Web databases on a given topic by selecting links within a topic which are more probable to lead to webpages that contain searchable web forms. They also proposed adaptive crawler [15] to automatically learn patterns of promising links and to adapt topic focus of the crawl as it crawl more links.

Sriram Raghavan and Hector Garcia-Molina [3] proposed a model of a hidden Web crawler for extracting contents from hidden web. Similarly, Jayant Madhavan et al. [16] proposed a system for surfacing Deep Web content, i.e. pre-evaluating submissions for each HTML form and adding the resulting HTML webpages into a search engine index.

Our work relating to data extraction from the Deep Web is different to above mentioned techniques, in such a way that it is directly passed user query to deep websites and extract data related to user query, while, other developed systems overlooked about automating query generation and sometimes extract irrelevant data by submitting query into web form interfaces.

There are two main steps in developing a system like DWX:

- **Resource Discovery:**

  In resource discovery, the task is to identify websites and databases that are expected to be relevant to the task.

- **Content Extraction:**

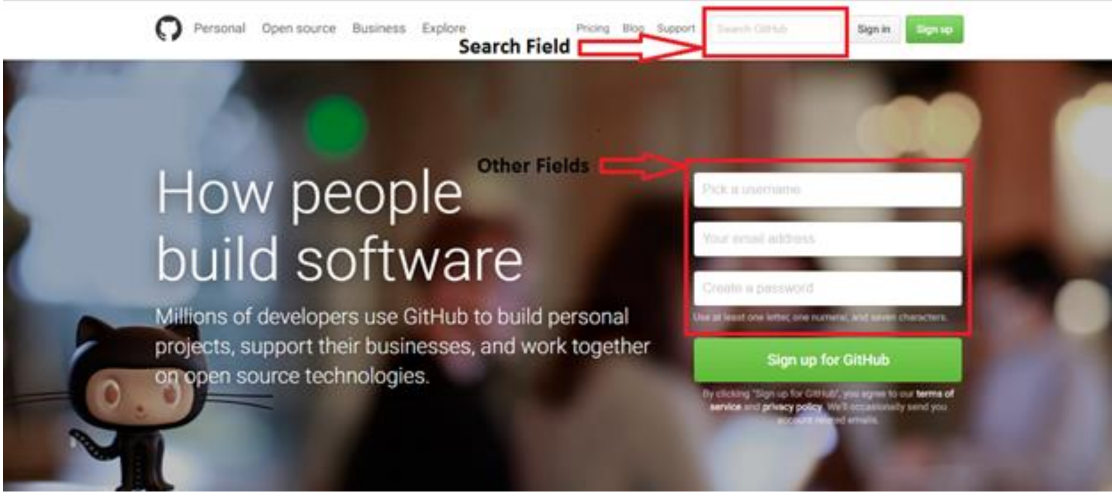
  In content extraction, the task is to visit the identified sites to submit queries and extract data form webpages.

In this thesis, we do not directly address the resource discovery problem. The related work [17-19] describes the resource discovery problem (i.e., identifying websites and webpages relevant to a specific task or topic). Our work leverages existing resource discovery that is used to identify potential websites for the Deep Web resources. So, we focus on content extraction task.

2.4 Searchable Web Forms

The Deep Web, i.e., content hidden behind HTML forms, has been regarded as a significant gap in search engine coverage. It represents a large portion of the structured data on the web behind these forms. Therefore, retrieving Deep Web content has been a long standing challenge for the database community for several years. A significant and large growing amount of data is retrieved only by filling out HTML forms to perform specific query on an underlying web data source.

Searchable web forms are categorized as forms in which a search query is passed to retrieve data from data resource. Figure 2-1 shows the searchable web form interface.



How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.

Personal
Open source
Business
Explore
Pricing
Blog
Support
Search GitHub
Sign in
Sign up
Search Field
Other Fields
Pick a username
Your email address
Create a password
Use at least one letter, one numeral, and seven characters.
Sign up for GitHub
By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

Figure 2-1: Searchable Form Field

So, searchable web forms are good resource of deep web because they contain numerous amounts of data behind them that is only accessible by directing search queries to databases.

2.5 Searchable Databases

Searchable databases are the most valuable resource of deep web. Every searchable web form directs a query to searchable database. Searchable databases are those databases which reside on World Wide Web and are only accessible by submitting the query through web form. So, standard search engines cannot handle the searchable database to dig out information by web form submission. Therefore, data inside searchable databases go into deep web category.

Chapter 3: Web Form Classification

Classification plays a vital role in many information management and retrieval tasks. Since webpages are semi-structured documents in HTML form, so, webpages classification is not only important, but distinguished from traditional text classification. Web form classification, also known as Web form categorization, is the process of assigning a Web form to one predefined category labels e.g. login, registration, contact form, etc. Classification is basically regarded as a supervised machine learning problem in which a set of labeled training data is used to train a classifier\predictor, which can be applied to predict the label of unknown\test examples. Web form classification can also assist to improve the quality of data retrieval. It can also help for building efficient focused crawlers or domain-specific search engines.

In web form classification, the task is to predict that whether a web form is searchable or not. If so, then identify the field of web form (i.e. searchable form contains search query field), because every searchable web form contains a searchable database that is valuable source of deep web.

For web form classification, Formasaurus [20] Python tool has been used. Formasaurus is a Python package that tells the type of an HTML form and its fields using machine learning. It can detect if a form is a login, search, registration, password recovery, join mailing list, contact, order form or something else, and also detect which field is a password field or a search query field, etc.

The following sections provide the detail of web forms classification task, in which we discuss about dataset, feature selection, classification models and their evaluation criteria.

3.1 Data Acquisition

The empirical evaluation has been done on 1418 web forms dataset [21]. It was originally collected form Alexa Top one million websites. The dataset has been split into training and testing data, in which 1064 web forms have selected for training data, while, 354 web forms have been used for test data.

Table 3-1 shows the distribution of different number of form’s element in dataset.

No. of form elements	Form Type	Form Label
415	Search	S
246	Login	L
165	Registration	R
143	Other	O
138	Contact	C
132	Join mailing list	M
105	Password / login recovery	P
74	Order / add to cart	B
Total Form: 1418		

Table 3-1: Labeled HTML Forms

3.2 Features Selection

3.2.1 Form Type Detection

Classifier is fed with a form which contains raw HTML. Using different classification models such as Logistic Regression, SGD and SVM we classify its type as shown in Table 3-1. The prominent features for form type detection are given as follow:

- Counts of the number of <form> tag element
- Form method (POST or GET)
- Presence of text on submit buttons
- Names and char n-grams of CSS classes and IDs
- Input labels
- Existence of specific substrings in URLs
- etc.

In case of searchable form detection, the useful features are given below:

- A single search query field
- A search query field named "q" or "s" or something else
- Presence of "search" word in URL
- Presence of "search" word in submit button text

- Presence of "search" word in form CSS Class or ID
- Form method whether is a GET or POST
- etc.

3.2.2 Field Type Detection

Conditional Random Field (CRF) model has been used to detect form field types, because field type has structured output e.g. in case of registration form, the possible fields are user’s name, email, password, etc. All fields in an HTML form is a sequence, where order is important. CRF allows taking field order in account. The useful features included for field type detection are given below:

- Form type predicted by a form type classifier
- Field tag name
- Field value
- Text before and after field
- Field CSS class and ID
- Text of field label element
- Field title and placeholder attributes
- etc.

3.3 Machine Learning Models and Evaluation

Logistic regression, SGD and SVM have been applied for web forms classification task, while, CRF has been applied for field type detection. These models have been trained on 1000 plus training examples and evaluated on 10 Fold cross validation.

3.3.1 Logistic Regression Classifier

Using Logistic Regression classifier, 88% of web forms have been classified correctly.

Table 3-2 shows the classification report of form type detection including Precision, Recall and f_1-score on 10-Fold cross validation using Logistic Regression classifier.

Form type	Precision	Recall	f_1-score	Support
Search	0.90	0.96	0.93	415
Login	0.96	0.95	0.95	246

Registration	0.93	0.87	0.90	165
Password / login recovery	0.84	0.81	0.83	105
Join mailing list	0.87	0.87	0.87	132
Contact / comment	0.85	0.93	0.89	138
Other	0.66	0.68	0.67	143
Order / add to cart	0.96	0.61	0.74	74
Avg. / Total	0.88	0.88	0.88	1418

Table 3-2: Classification Report of Form type detection on 10-Fold cross validation using Logistic Regression Classifier

3.3.2 SGD Classifier

Using SGD classifier, 88% of web forms have been classified correctly.

Table 3-3 shows the classification report of form type detection including Precision, Recall and f_1-score on 10-Fold cross validation using SGD classifier.

Form type	**Precision**	**Recall**	**f_1-score**	**Support**
Search	0.90	0.96	0.93	415
Login	0.96	0.95	0.96	246
Registration	0.94	0.85	0.90	165
Password / login recovery	0.80	0.82	0.81	105
Join mailing list	0.89	0.86	0.88	132
Contact / comment	0.87	0.93	0.90	138
Other	0.65	0.71	0.68	143
Order / add to cart	0.98	0.59	0.74	74
Avg. / Total	0.88	0.88	0.88	1418

Table 3-3: Classification Report of Form type detection on 10-Fold cross validation using SGD Classifier

3.3.3 Linear SVM Classifier

Using Linear SVM classifier, 89% of web forms have been classified correctly.

Table 3-4 shows the classification report of form type detection including Precision, Recall and f_1-score on 10-Fold cross validation using Linear SVM classifier.

Form type	Precision	Recall	f_1-score	Support
Search	0.92	0.96	0.94	415
Login	0.97	0.96	0.96	246
Registration	0.94	0.90	0.92	165
Password / login recovery	0.85	0.82	0.83	105
Join mailing list	0.88	0.86	0.87	132
Contact / comment	0.85	0.93	0.89	138
Other	0.67	0.71	0.69	143
Order / add to cart	0.94	0.62	0.75	74
Avg. / Total	0.89	0.89	0.89	1418

Table 3-4: Classification Report of Form type detection on 10-Fold cross validation using Linear SVM Classifier

3.3.4 CRF Classifier

Using CRF classifier, 86% of web forms fields have been classified correctly.

Table 3-5 shows the classification report of field type detection including Precision, Recall and f_1-score on 10-Fold cross validation using CRF (Conditional Random Field) classifier.

Field type	Precision	Recall	f_1-score	Support
Search query	0.843	0.980	0.907	99
Email	0.945	0.987	0.966	156
Password	1.000	0.966	0.983	88
Product quantity	1.000	0.875	0.933	8
Submit button	0.895	1.000	0.944	68
Username	0.767	0.767	0.767	43
Password confirmation	1.000	1.000	1.000	24
Receive emails confirmation	0.909	0.370	0.526	27
First name	0.913	0.840	0.875	25
Last name	0.870	0.800	0.833	25
Organization name	1.000	0.417	0.588	12
Address	0.706	0.667	0.686	18

City	0.909	0.714	0.800	14
State	1.000	0.750	0.857	4
Postal code	1.000	0.929	0.963	14
Country	0.875	0.636	0.737	11
Phone	1.000	0.944	0.971	18
Fax	1.000	1.000	1.000	1
TOS confirmation	1.000	0.692	0.818	13
Comment text	0.786	0.971	0.868	34
Captcha	0.962	0.735	0.833	34
Remember me checkbox	1.000	1.000	1.000	29
Username or email	0.667	0.222	0.333	9
Other	0.730	0.854	0.787	171
Full name	0.595	0.926	0.725	27
Search category / refinement	0.842	0.985	0.908	65
Other read-only	1.000	0.500	0.667	6
Style select	1.000	1.000	1.000	6
Email confirmation	1.000	1.000	1.000	6
Time zone	1.000	0.667	0.800	3
DST	1.000	1.000	1.000	2
Gender	1.000	0.958	0.979	24
About me text	0.000	0.000	0.000	3
Reset / clear button	1.000	1.000	1.000	3
Security question	0.000	0.000	0.000	0
Answer to security question	0.000	0.000	0.000	0
Comment title or subject	0.773	0.810	0.791	21
Full date	0.571	0.444	0.500	9
Year	1.000	0.875	0.933	8
URL	1.000	0.400	0.571	10
Cancel button	0.000	0.000	0.000	0
Sorting option	0.833	0.455	0.588	11
Middle name	0.000	0.000	0.000	0

Day	1.000	0.833	0.909	6
Month	1.000	0.833	0.909	6
Honeypot	0.333	0.100	0.154	10
Other number	1.000	0.200	0.333	10
OpenID	1.000	1.000	1.000	1
Avg. / Total	0.868	0.860	0.850	1182

Table 3-5: Classification Report of Field type detection on 10-Fold cross validation using CRF Classifier

3.4 Results Comparison

The empirical results in Table 3-6 revealed that Linear SVM classifier has better precision, recall and f_1-score than that of other two classifiers (i.e. Logistic Regression and SGD). So, form type detection has been carried out by Linear SVM, while, form field type detection has been done with CRF (Conditional Random Field).

Evaluation Measure \ Classification Models	**Logistic Regression**	**Stochastic Gradient Descent (SGD)**	**Linear SVM**
Precision	0.88	0.88	0.89
Recall	0.88	0.88	0.89
f_1-score	0.88	0.88	0.89

Table 3-6: Precision, Recall and f_1-score of LR, SGD and SVM

Chapter 4: Application Architecture and Implementation

4.1 Application Modules

The proposed application architecture of the DWX system consists of the following major components as shown in Figure 5-1:

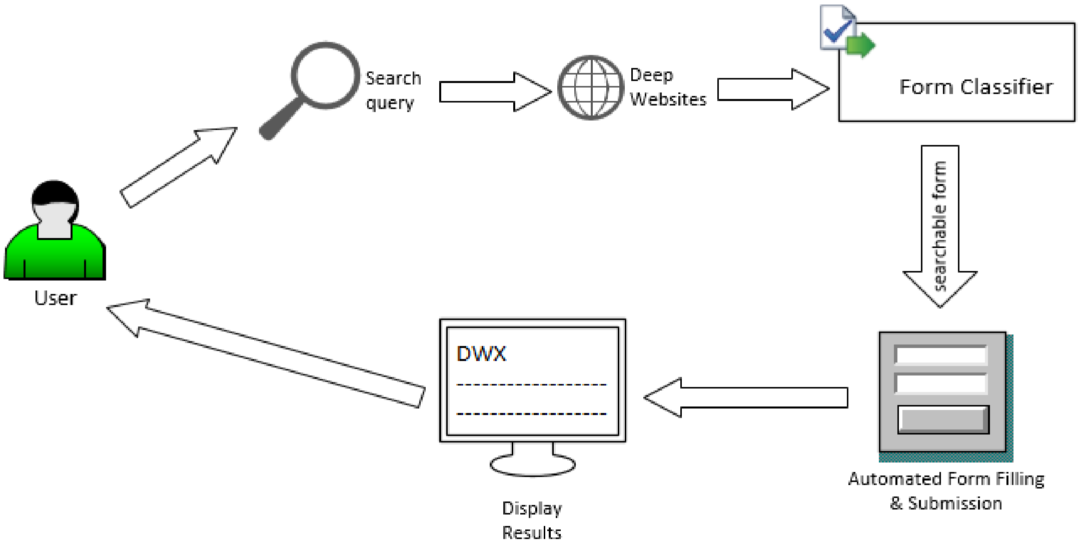


Figure 4-1: DWX Application Architecture

DWX application comprises of different modules in which each module performs its work. Now we elaborate each of the major components:

4.1.1 User Interface

It presents a high level interface to the user for passing queries. Queries are then submitted to the deep web resources corresponding to specified queries.

Figure 4-2 shows search query interface for DWX system.

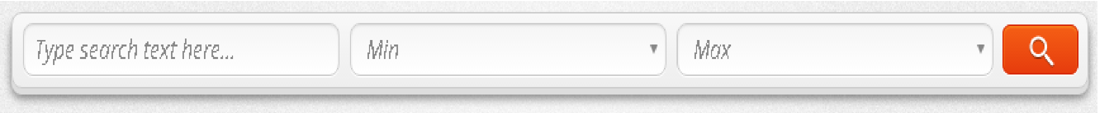


Figure 4-2: Search Query Interface

4.1.2 Form Classifier

Form classifier detects the searchable form in the website that directs query to searchable databases. If the form is a searchable, it will identify the fields in that form. Forms classifier run on the server and doesn't contribute in the response time of end user.

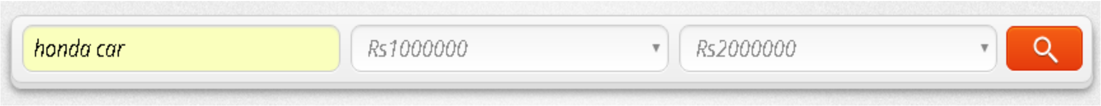
4.1.3 Automated Form Filling & Submission

The identified searchable form fields from the Form Classifier have been automatically filled at server with user specified query and submitted to searchable databases. This entire process is hidden form user.

4.1.4 Response Results

The response from searchable databases displayed to the user who initiated the search query.

For example, assume user has passed the query as shown below in Figure 4-3:



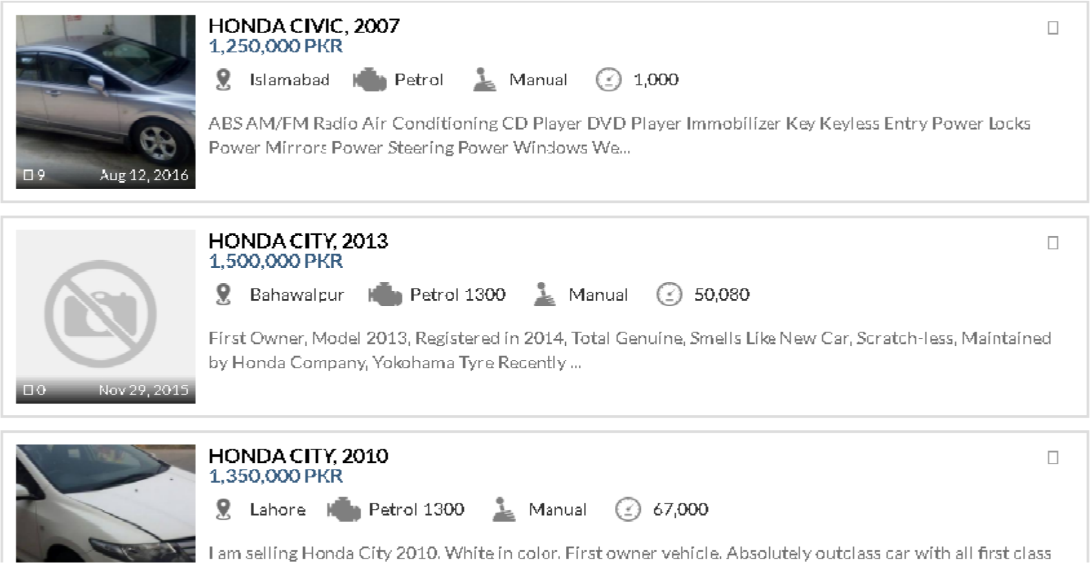
honda car

Rs1000000 ▾

Rs2000000 ▾

Figure 4-3: Search Query (Honda car, price range b/w 1000000 to 2000000)

The corresponding response results displayed to user as shown given below in Figure 4-4:



HONDA CIVIC, 2007
1,250,000 PKR
Islamabad Petrol Manual 1,000
ABS AM/FM Radio Air Conditioning CD Player DVD Player Immobilizer Key Keyless Entry Power Locks Power Mirrors Power Steering Power Windows We...
9 Aug 12, 2016

HONDA CITY, 2013
1,500,000 PKR
Bahawalpur Petrol 1300 Manual 50,080
First Owner, Model 2013, Registered in 2014, Total Genuine, Smells Like New Car, Scratch-less, Maintained by Honda Company, Yokohama Tyre Recently ...
0 Nov 29, 2015

HONDA CITY, 2010
1,350,000 PKR
Lahore Petrol 1300 Manual 67,000
I am selling Honda City 2010. White in color. First owner vehicle. Absolutely outclass car with all first class

Figure 4-4: Query Response Results

4.2 Front-End Handling

The front-end of DWX comprised of user interface module and response results module. The user interface and response results modules have been developed in HTML and CSS. The user interface contained a web filled by the user for further processing, while, response results module consists of a simply displaying response web page.

4.3 Back-End Handling

The back-end of DWX contains form classifier module, automated form filling and submission module. The back-end of DWX system handled the request coming from front-end, which is developed in Flask [22] and Python. Flask is basically a web application framework for python. Form classifier module is written in Python (a programming language) which used the "scikit-learn [23]" machine learning library, while, automated form filling and submission module has written in Python programming, which used "urllib [24]" and "urllib2 [25]" python libraries for handling HTML request and response.

Chapter 5: Large Scale Data Extraction

Since user query has been directed to multiple deep websites in the sequential manner, so, each response per website takes 1 to 2 second. If user query is submitted to 100 websites it will take 100 seconds to return the response against one user query. That is much larger response time for user. Therefore, there is a need to make the query response time as fast as possible. Faster query response can be achieved by distributed processing of the query against each website the query.

5.1 Large Scale Data Extraction

To extract data from Deep Web resources, user queries as well as deep websites URLs have been distributed across multiple worker machines by the master machine. This task has been carried out by Apache Spark [26]. Apache Spark is an open source big data processing framework that is built for faster speed, ease of use, and data analytics. It works like Hadoop “MapReduce” framework in which Map and Reduce tasks are distributed on cluster machines.

Each worker locally runs Form Classifier module and Automated Form Filling and Submission module. When user submits his search query, this query has been passed into multiple deep websites. The master machine distributes user query and websites URLs across multiple workers. Each worker identifies Form type from deep website and automatically fill that form and submit it to corresponding searchable data base. The response pages form each worker machine returned to master machine that concatenates these responses into a single display result.

Figure 5-1 shows the distribution of queries and deep websites URLs across multiple worker machines by the master machine.

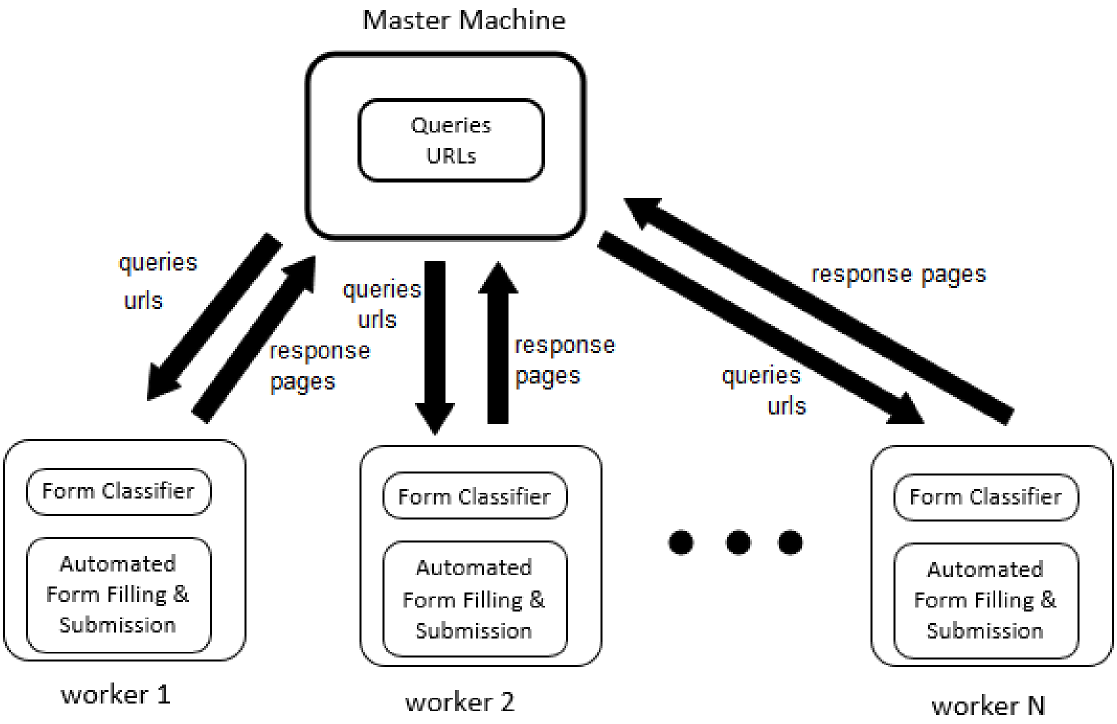


Figure 5-1: Distributing Queries and URLs on worker machines

Chapter 6: Conclusion and Future work

Massive and high quality data from the Deep Web resource is useful to build a knowledge based databases (i.e. structured relational databases). Exploring and extracting the vast deep web content tends to a major research challenge for knowledge discovery and information retrieval community. In this thesis, an efficient DWX system has been developed for extracting the contents from deep web resources. The DWX system heavily relies on “form classifier”, and “automated form filling and submission” modules. Form classifier module identifies the searchable web interfaces in the deep websites, while, automated form filling and submission module automatically fills the searchable forms fields with query. The empirical results show that the DWX system is robust against changes in “form interfaces” and their fields.

The under discussion form classifier is specifically capable of detecting web form type in which <*form*> tag element is present. In modern era, web form is created in JavaScript in which all action can be done in JavaScript function (e.g. onClick()) instead of <*form*> *tag* element. So, form classifier is lacking to detect such type of form based on JavaScript. Therefore, there is still a room to enhance the capability of “form classifier” by detecting the JavaScript based forms in future.

References

[1] N. Hamilton, "The Mechanics of a Deep Net Metasearch Engine," in *WWW*, 2003.

[2] J. Devine and F. Egger-Sider, "Beyond Google: the invisible web in the academic library," *The Journal of Academic Librarianship,* vol. 30, pp. 265-269, 2004.

[3] S. Raghavan and H. Garcia-Molina, "Crawling the hidden web," 2000.

[4] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: a survey," *Knowledge-based systems,* vol. 70, pp. 301-323, 2014.

[5] S. Pederson. (2003, Understanding the Deep Web in 10 minutes.

[6] A. Bosch, T. Bogers, and M. Kunder, "Estimating search engine index size variability: a 9-year longitudinal study," *Scientometrics,* vol. 107, pp. 839-856, 2016.

[7] M. K. Bergman, "White paper: the deep web: surfacing hidden value," *Journal of electronic publishing,* vol. 7, 2001.

[8] (2016). *Deep web*. Available: https://en.wikipedia.org/wiki/Deep_web

[9] D. Wiener-Bronner. (2015, NASA is indexing the 'Deep Web' to show mankind what Google won't.

[10] J. Caverlee, L. Liu, and D. Buttler, "Probe, cluster, and discover: Focused extraction of qa-pagelets from the deep web," in *Data Engineering, 2004. Proceedings. 20th International Conference on*, pp. 103-114.

[11] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards automatic data extraction from large web sites," in *VLDB*, 2001, pp. 109-118.

[12] A. Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003, pp. 337-348.

[13] P. G. Ipeirotis and L. Gravano, "Distributed search over the hidden web: Hierarchical database sampling and selection," in *Proceedings of the 28th international conference on Very Large Data Bases*, 2002, pp. 394-405.

[14] L. Barbosa and J. Freire, "Searching for Hidden-Web Databases."

[15] L. Barbosa and J. Freire, "An adaptive crawler for locating hidden-web entry points," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 441-450.

[16] J. Madhavan, D. Ko, Ł. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's deep web crawl," *Proceedings of the VLDB Endowment,* vol. 1, pp. 1241-1252, 2008.

[17] S. Chakrabarti, M. Van den Berg, and B. Dom, "Focused crawling: a new approach to topic-specific Web resource discovery," *Computer Networks,* vol. 31, pp. 1623-1640, 1999.

[18] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, "Focused Crawling Using Context Graphs."

[19] A. McCallumzy, K. Nigamy, J. Renniey, and K. Seymorey, "Building domain-specific search engines with machine learning techniques."

[20] (2016). *Formasaurus*. Available: https://pypi.python.org/pypi/formasaurus

[21] (2016), Web Forms Dataset. Available: https://github.com/TeamHG-Memex/Formasaurus/tree/master/formasaurus/data

[22] *Flask*. Available: http://flask.pocoo.org/

[23] *Scikit-learn*. Available: http://scikit-learn.org/stable/

[24] *Urllib*. Available: https://docs.python.org/2/library/urllib.html

[25] *Urllib2*. Available: https://docs.python.org/2/howto/urllib2.html

[26] *Apache Spark*. Available: http://spark.apache.org/

About the Author

Rao Muhammad Umer was born in Sahiwal, Punjab, Pakistan on 11th April 1992. He did his matriculation from Lasani Public High School, Sahiwal and his FSc. from Govt. Post Graduate College, Sahiwal. His undergraduate degree was in BSc. Computer Systems Engineering form The Islamia University of Bahawalpur in 2010 under fully funded scholarship of National ICT R & D Fund, sponsored by Federal Govt. of Pakistan. After the completion of his undergraduate degree with distinction, he joined in the MS program in Computer Science at PIEAS in 2014 under PIEAS IT-Endowment scholarship, sponsored by Higher Education Commission of Pakistan. His research interests include Data Science, Machine Learning, Deep Learning, and High Performance Computing.

Contact Information:

Department of Computer & Information Sciences (DCIS),
Pakistan Institute of Engineering and Applied Sciences (PIEAS),
PO Box Nilore 45650,
Islamabad, Pakistan.

Cell: +92-336-7675246
Website: http://raoumer.github.io
Email: engr.raoumer943@gmail.com