DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.4936

ISSN No. 0976-5697

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

GENERATING QUERIES TO CRAWL HIDDEN WEB USING KEYWORD SAMPLING AND RANDOM FOREST CLASSIFIER

Sabarni Kundu
Electronics and Communication Engineering
Maharaja Surajmal Institute of Technology
New Delhi

Shwetanshu Rohatgi
Computer Science and Engineering
Maharaja Surajmal Institute of Technology
New Delhi, India

Abstract: One of the most challenging aspects in information retrieval systems is to crawl and index deep web. A deep web is part of World Wide Web which is not visible publically and therefore can't be indexed. There is a huge amount of scholarly data, images and videos available in deep web which if indexed can serve purpose of research and stop illegal activities. We propose an efficient hidden web crawler that uses Sampling and Associativity Rules in order to find the most important and relevant keywords which are used to generate queries that can extract information from databases and web forms. Further, we use random forest technique to index out search results. Our web crawler has capabilities to efficiently overcome various prior challenges that we have stated in this paper.

Keywords: **Deep Web**; Dark Web; Crawler; Random Forest Classifier; Apriori Intuition; Keyword Sampling; TF-IDF; NLP; Database Querying

I. INTRODUCTION

WW or World Wide Web is defined as "wide-area Whypermedia information retrieval initiative aiming to give universal access to large universe of documents" [1].

Recently, the usage of internet is multiplying rapidly. WWW has been known for proving a vast source of information. Due to its rapidly increasing usage we need to design an effective search engine .Web crawlers are the intrinsic part of search engine that provides growing of web pages in methodical and automated manner or in orderly fashion [2].

Web crawling or spidering is basically a process where we amass web pages from internet. The basic program of web crawler is to traverse the internet automatically by downloading links from one web page to another web page and so forth. It is the best tool where we can collect the web pages and index them to and successively keep our database updated.

Deep web is usually referred as part of WWW that is not visible publically and hidden under surface web. In deep web, the pages are not indexed or queued by standard search engines, therefore the content on hidden web or deep web is difficult to accessible. The data that we fetch from hidden web is strutted one and indexing technique implemented for structured and unstructured data is completely different [3].

In this paper we will be discussing the basic working and principles of web crawler and further will be briefing about deep web and dark web along with their crawlers. Section 1 is devoted to the architecture of web crawler used in surface web. Section 2 will consist brief about Deep web. Section 3 will mainly focus on the Deep web crawler and its architecture. Section 4 will be about Dark web. Section 5 will be about searching technique implemented in Dark web. Section 6 will be about our Challenges faced by Deep web crawlers by studying previous works. In section 7 we proposed Hidden web crawler whereas section 8 will demonstrate experimental results. In the last section we conclude our paper with future scope and references.

II. SURFACE CRAWLING

The main purpose of web crawler is to fetch URL and download the corresponding pages mention in the webpage. Web crawlers are essential part of search engine where they amass the corpus of webpages queued by the engine itself.

Initially web crawler starts its system by setting of URL request. All the important URLs that are to be retrieved and given priority are kept in URL queue and from here the crawler gets a URL link and download the corresponding webpage. After page downloading URLs are passed to the extractor which would extract the required data given by the users and then data can be organized into groups and further URL can be pushed back to queue. This process is repeated over and over again till the URL queue is empty [1].

A. Architecture of Surface Web Crawler

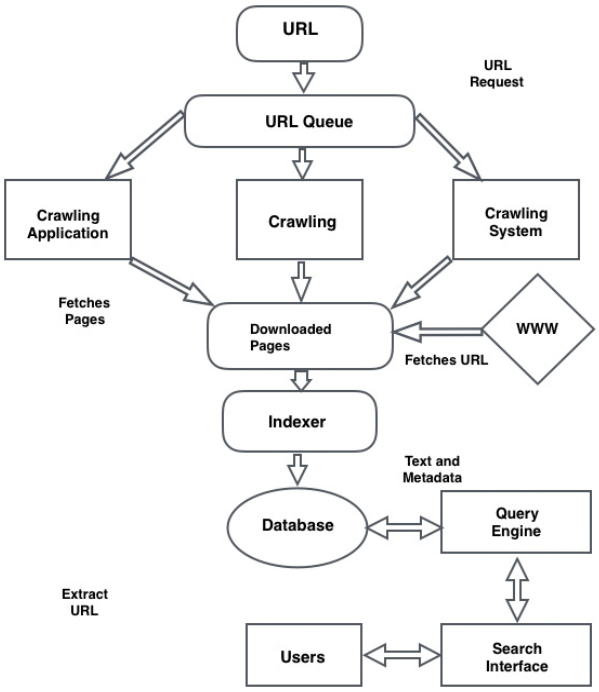

Figure 1. Web Crawler Architecture.

III. DEEP WEB

Deep web is growing exponentially and at a rate that defies quantification. It's almost impossible to measure the size of deep web, recently it has been estimated that its about 4000-5000 times bigger than surface web. The contents of Deep Web are hidden from standard search engine as they require a query to produce results. These websites may have 100 of pages to navigate through but 1000s of pages can be searched. Let's take an example of famous news channel where we can visit the web pages but cannot fetch the databases.

Deep web is a complex process and it is classified into 2 categories of data [4].

Category 1 involves all the details or web pages that are difficult to fetch through standard search engine, these pages can involve Facebook or twitter posts, webpages that are buried under many layers down in dynamic pages. It also involves the result that sits so far down the standard search that normal users will never find them.

Category 2 involves a vast repository of information that is not accessible to standard search engine. This consist of information found in webpages, databases and many other sources. It can be only browsed through custom query, which cannot be done by the standard search engine used in surface web.

Deep Web consist of both structured and unstructured content. This information is compiled by experts, researchers through automated processing system. Deep Web connections are anonymous and hard to make a check of, facilitating access to illegal information and resources from around the world without government filtering, "interpretation" or censorship.

IV. DEEP WEB CRAWLER

A. Architecture

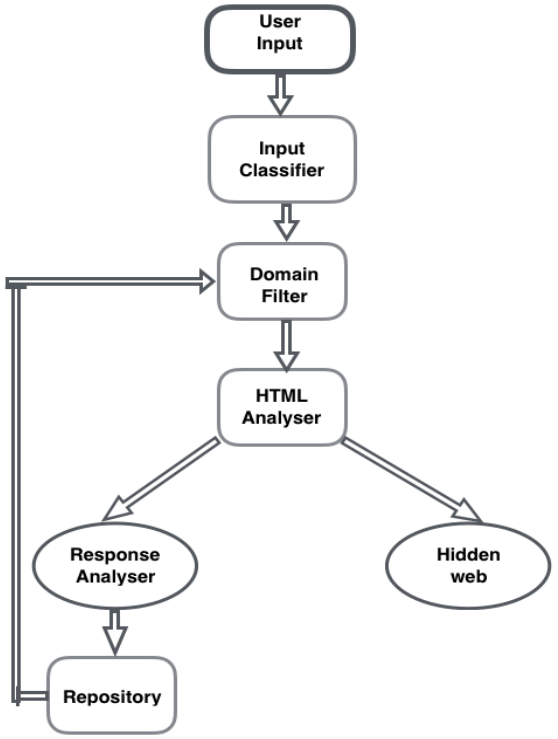

Figure. 2. Deep Web Crawler Architecture.

V. DARK WEB

The Surface Web is anything that can be indexed by a typical search engine like Google, Bing or Yahoo and deep web is anything that a standard search engine can't access or indexed [5]. The Dark Web then is classified as a small portion of the Deep Web that has been intentionally hidden and is inaccessible through standard web browsers. The major portion of the data that makes up the Dark Web resides on an anonymous Internet known as the TOR network. The TOR network is an anonymous network that can only be accessed with a special web browser, called the TOR browser. This is the portion of the Internet most widely known for illicit activities because of the anonymity associated with the TOR network.

VI. CRAWLING HIDDEN WEB

Millions of web pages are crawled and queued daily by searching through endless hyperlinks. Yet a large amount of data is hidden behind the web queries. The information of web content is behind web forms and the client side scripting is referred to as the hidden web, which is estimated to consist of many millions of web pages.

Deep web or Hidden Web consist of a dynamically generated internet pages which is not accessed by standard search engines, we need to access these by creating a query in a deep web and thus fetching relevant information. Our main aim is to crawl relevant parts of the hidden web and thus fetch information related to our demands and needs.

While crawling deep web we usually take three parameters as our input parameter and those are set of seed URLs, User data and Domain specific data as shown in Fig. 2.

Input-classifier then selects the web page elements and after that a domain filter uses this data and fill up the html/web forms and thus passes updated result to the analyser, then the analyser submits the form to the web server and fetches the nascent web formed and according to it our database gets edited and in this way this process is iterated over till crawl capacity.

VII. PRIOIR RESEARCH AND CHALLENGES

A. Various Challenges

In [6] we can see an effective HiWE model where the crawler first built an internal representation of searches and then further representation in a vector form. Further the match algorithm compares the internal form representation and current contents and hence assigning a value assignment and then the response is stored in the repository.

In [7] A. Bergholz, B. Chidlovskii have highlighted a system for domain specific crawling for the hidden web buts this crawler is only for full text search form. These forms searches any web documents only through single text field which indicate a full text. They generate a problem of keyword query when it crawl all the contents behind a web.

In [8] S. Liddle, D. Embley, Del Scott and S. Ho Yau, proposed a framework to extract data from hidden web forms. It represents the problem of extracting the full contents behind a web forms. Due to this problem this system does not accept forms with the required “textbox” fields to be filled in.

In [9] the system is very efficient as it automatically generates new queries from the result of the previous queries but in this crawler the system is not properly indexed.

VIII. PROPOSED HIDDEN WEB CRAWLER

A. Proposed Theory

Instead of crawling the full content of the web we can crawl some selective content that will make our system more efficient and saves our time. For designing an effective crawling system we create such an algorithm that focuses on crawling only the necessary details and then creating a query based on crawl results.

Step1: Creating a corpus vector

In this step we extract out the unnecessary contents such as stopwords, punctuators, various symbols and white spaces using “tm” function of NLP and thus storing all these data into a vector.

Step 2: Clustering the different content

After creating a corpus vector we can then organise our contents into different clusters. First cluster that consist of the content of corpus vector and second cluster will consist of the remaining content of the website.

Step 3: Finding the most frequently occurred word or Keyword

We will be applying the sampling function and TF-IDF [10] function and thus determine the most frequently occurring word which will be our keyword.

Step 4 Apriori intuition

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. Apriori [8] uses a breadth-first search strategy to count the support of item sets and uses a candidate generation function which exploits the downward closure property of support.

We will be applying associative rule learning i.e. Apriori Intuition to our keyword that we have obtained in the previous steps. By this algorithm we can predict other contents or topics which will be related to our keywords and thus providing a user, a wide variety of web pages related to the searches [11].

Step 5: Generating Query

After fetching the appropriate keyword we then generate a query [12]. This will help us parse hidden databases and web forms and hence we successfully crawl a Deep web and webpage in an effective way.

B. URL Queuing Technique

Earlier we have used depth first search, breadth first search and best first search for URL ordering. Among these DFS is also used in crawling system such as Fish Search [13]. BSF (Breath first search) is considered as one of the easiest method for indexing, it worked well. But however BFS (Breath first Search) didn't produce a satisfactory in focused crawling [14]. However, Best First Search overcame these problems. Best First searches uses technique such as link analysis or text analysis or a combination of both for an effective result. Now in this text analysis we uses the concept of similarity scoring where we use machine learning algorithm. We can use similarity equation along with the contents and URL of the page. This procedure is quite effective, but we can produce much effective process of indexing in a focused crawler by applying few machine learning algorithm.

Thus for an effective indexing we can use Support Vector Machines, where with the help of space vector model and cosine similarity we can index the pages.

Furthermore we can also use genetic algorithm for URL for topic specific searches. These process are quite effective in focused crawling, but for correct and much more accurate results in focused crawling use Random Forest Intuition.

Random Forest is also known as Random Decision Forest and this is a part of ensemble learning which is used in classifiers and regressors [15]. This algorithm involves technique of bootstrap aggregating and because of this special property it produces the most accurate and effective ordering of pages as it mitigate the variance without increasing the bias. Further, we can make the uncertainty of the following prediction by the standard deviation of the predicted values.

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}.$$

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Dataset of the URLs to be crawled: X = x1…..xn
Responses: Y = y1……yn
Number of Samples: B

Our proposed model is used for querying the web links and the results produced are indexed and classified using Random Forest.

We use Random Forest primarily because a lot of Hidden web links and data beneath the hidden web has been in the form of images and videos than in the form of text and for images and videos Random forest has been found to be the most efficient and accurate algorithm.

In above equation we calculate Sigma that will help us to reduce variance without increasing bias that removes the problem of over fitting as in case of decision trees and this is what makes Random forest the go-to technique for classifying our data and web links.

C. Proposed Architecture for Hidden Web Crawler

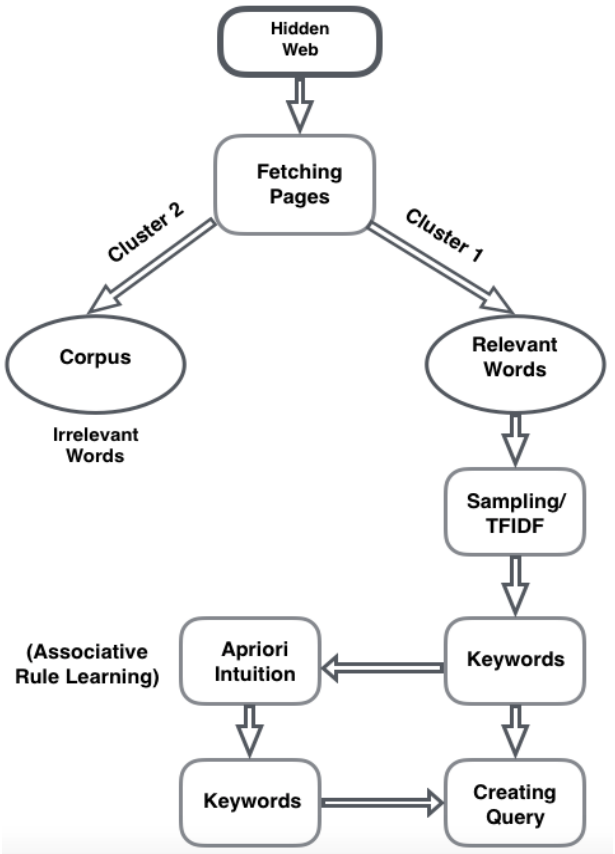

Figure 3. Proposed Hidden Web Crawler Architecture

IX. EXPERIMENTAL RESULTS

We ran our proposed web crawler on a set of text and database that has similar Structured and Unstructured data in various databases that forms the part of hidden web. Using R studio and NLP toolkit we analyzed the performance of our web crawler on database of over 20,000 web forms and database, we segregated over data into two clusters using 'tm' function in NLP library which produced a corpus vector and a relevant word cloud or vector. Using relevant words we performed sampling and TF-IDF [10] which is a general technique in information retrieval systems, used to find relevant and important keywords from a document.

Further we applied apriori algorithm to our keywords vector, which is based on association rule discovery including support, confidence i.e. "if-then rules". Our apriori algorithm gave us set of keywords that were related to our sampled keywords after TD-IDF step. This largely improved our word cloud as our crawler will now index keywords as well as related words similar to keywords produced by Apriori recommender engine. We limited the number of relevant keywords in our data set to 500 and we further produced queries from these keywords.

We ran our produced queries on World Wide Web and we could come up with 7836 new URLs that had not been previously crawled by any of the search engine including Google, Bing and Yahoo. These are hidden web page links. Our page links were rich in image and video content that had not been indexed properly. Out of 7836 our 73% links had no back links meaning they had not been properly linked via hypertext due to which they were never crawled.

Our crawled links had been majorly dominated by Images and Videos that has not been labelled and segmented. We used our Random Forrest Classifier and various other already used classification techniques like BFS, DFS SVMs etc. Our model produced the best results with Random Forest Classifier and produced least Turnaround time as shown in Fig. 4. Hence we experimentally infer that Random tree are 2.3 times more accurate than BFS and DFS techniques and about 70% more efficient than Support Vector Machine (SVMs)

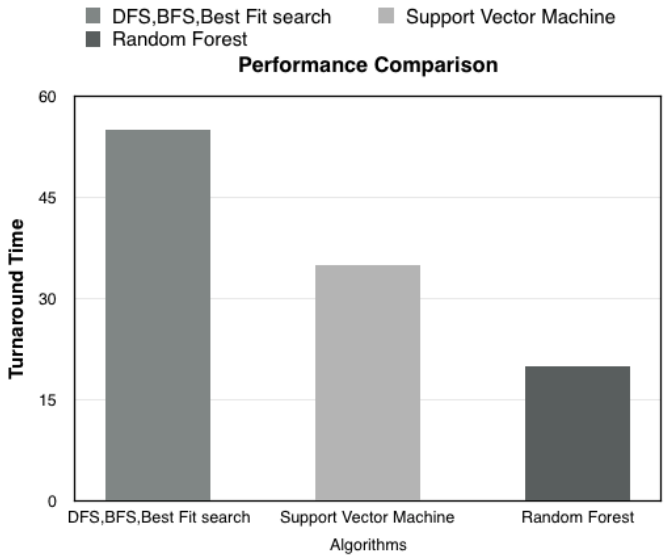

Figure 4. Performance Comparison for indexing the crawled links.

We also used decision trees but they were vastly slow in classification of unlabeled images and videos and since our Hidden web crawled results were primarily as Images and

Videos, we used Random forest classification which works the best for Images and Videos classification on hidden web.

X. END SECTION

A. FUTURE SCOPE

These is a lot that needs to be done in order to index deep web as well as dark web. Deep has opened a wide range of opportunities for scholars by indexing various research papers and articles but also forms a part of illegal activities that happen in the dark web in the form selling personal information, drugs etc. If this part of the web is index like surface web by devising dynamic contiguously converging and efficient web crawling techniques then such illegal activities can be put to an end.

B. CONCLUSION

We have demonstrated both theoretically and experimentally that our proposed crawler has the capabilities to crawl hidden web and extract data from web forms effectively by using generated queries and then indexing them using Random forest classifier. We have tested our query generator on a dataset of words using NLP toolkit and our classifier is more efficient in indexing pages on hidden web for both structured and unstructured data specially classification of images and videos. Our web crawler has successfully overcome prior challenges that we demonstrated in section 7 and we look forward to make hidden web much more accessible and safe for scholars and researchers

XI. REFERENCES

[1] A comparative study on web crawling for searching hidden web by IJCSIT

[2] Trupti V. Udapure, Ravindra D. Kale and Rajesh C. Dharmik,"Study of web crawler and its Different types", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 01-05

[3] Ali Mesbah , Arie van Deursen , Stefan Lenselink, Crawling Ajax-Based Web Applications through Dynamic Analysis of User Interface State Changes, ACM Transactions on the Web (TWEB), v.6 n.1, p.1-30, March 2012

[4] BERGMAN, M. 2000. The deep Web: Surfacing the hidden value. BrightPlanet, www.completeplanet.com/Tutorials/DeepWeb/index.asp.

[5] BERGMAN, M. 2000. The deep Web: Surfacing the hidden value. BrightPlanet, https://brightplanet.com/2014/03/clearing-confusion-deep-web-vs-dark-web.asp

[6] C. J. Kaufman, Rocky Mountain Research Laboratories, Boulder, Colo., personal

[7] communication, 1992. (Personal communication)

[8] A. Bergholz, B. Chidlovskii, "Crawling for Domain-Specific Hidden Web Resources" In the Proc. of the 4th Int. Conf. on Web Information System Engineering,2003

[9] S. Liddle, D. Embley, Del Scott and S. Ho Yau, " Extracting Data Behind Web Forms" In the Proc. of the 28th Int. Conf. on Very Large Data Bases, China, 2005

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

[11] LUO Xin; XIA De-lin; YAN Pu-liu. Improved feature selection method and TF-IDF formula based on word frequency differentia. Computer Applications, 2005, 25(9): 2031-2033.

[12] Markus Hegland. The Apriori Algorithm – a Tutorial. CMA, Australian National University, WSPC/Lecture Notes Series, 22-27. March 30, 2005.

[13] L. Barbosa and J. Freire, "Siphoning hidden-web data through keyword-based interfaces," in Proceedings of the 19th Brazilian Symposium on Databases SBBD, 2004.

[14] Cho, J., Garcia-Molina, H., & Page, L. (1998). Efficient crawling through URL ordering. Computer Networks and ISDN Systems, 30(1–7), 161–172.

[15] De Bra, P.M.E. & Post, R.D.J. (1994). Information retrieval in the World- Wide Web: Making client-based searching feasible. In Proceedings of the First World-Wide Web Conference (pp. 183–192). New York: ACM Press.

[16] L. Breiman. Random forests. Machine learning, 45(1):5–32, 2001.