Deep Web Entity Monitoring

Mohammadreza Khelghati
Database Group
University of Twente, Netherlands
s.m.khelghati@utwente.nl

Djoerd Hiemstra
Database Group
University of Twente, Netherlands
d.hiemstra@utwente.nl

Maurice van Keulen
Database Group
University of Twente, Netherlands
m.vankeulen@utwente.nl

Categories and Subject Descriptors

H3 [**INFORMATION STORAGE AND RETRIEVAL**]: [Information Search and Retrieval]

General Terms

Design, Experimentation, Performance

Keywords

Crawling, Deep Web, Entity Monitoring, Web Harvesting

1. INTRODUCTION

Accessing information is an essential factor in decision making processes occurring in different domains. Therefore, broadening the coverage of available information for the decision makers is of a vital importance. In such a information-thirsty environment, accessing every source of information is considered highly valuable. Nowadays, the main or the most general approach for finding and accessing information sources is searching queries over general search engines such as Google, Yahoo, or Bing. However, these search engines do not cover all the data available on the Web. In addition to the fact that none of these search engines cover all the webpages existing on the Web, they miss the data behind web search forms. This data is defined as *hidden web or deep web* which is not accessible through search engines. It is estimated that deep web contains data in a scale several times bigger than the data accessible through search engines which is referred to as *surface web* [9, 6]. Although this information on deep web could be accessed through their own interfaces, finding and querying all the interesting sources of information that might be useful could be a difficult, time-consuming and tiring task. Considering the huge amount of information that might be related to one's information needs, it might be even impossible for a person to cover all the deep web sources of his interest. Therefore, there is a great demand for applications which can facilitate accessing this big amount of data being locked behind web search forms. Realizing approaches to meet this demand is one of the main issues targeted in this PhD project. Having provided the access to deep web data, different techniques can be applied to provide users with additional values out of this data. Analyzing data, finding patterns and relationships among different data items and also data sources are considered as some of these techniques. However, in this research, monitoring entities existing in deep web sources is targeted.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

1.1 What is Deep Web?

For discovering the content of the Web, the most applied method by search engines is to rely on following every link on a visited page [1, 8]. The content of these pages would be analyzed and indexed for being matched against user queries later. This content is referred as *surface web* [1]. This way of accessing web pages makes part of the Web unavailable for the search engines; the part which is hidden behind web forms. In order to access this data, one should submit queries through forms provided by each web source. This is illustrated in Figure 1.1. As this part of the Web is invisible or hidden for the search engines, it is called *hidden* or *invisible web* [1]. However, by applying a series of techniques, the invisible web could be accessible to users. Therefore, it is also called as *deep web*. Deep web refers to the content hidden behind web forms through which standard crawl techniques cannot easily access data [5, 1, 7].

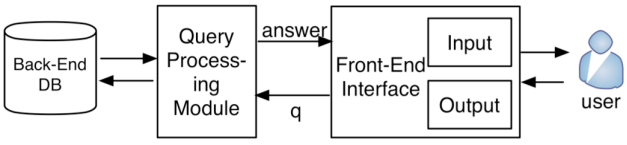

Figure 1: Accessing Deep Website by User [12]

The deep web could include dynamic, unlinked, limited access, scripted, or non-HTML/text content residing in domain specific databases, and private or contextual web [1]. This content could exist as structured, unstructured, or semi-structured data in searchable data sources. The results from these data sources can only be discovered by a direct query. The deep web content is diversely distributed across all subject areas from financial information, and shopping catalogs to flight schedules, and medical research [2].

1.2 Motivations on Targeting Deep Web Data

In the beginning of this section, the most important reason on accessing deep web data was mentioned. However, accessing more data sources is not the only reason which makes deep web data interesting for users, companies and therefore for researchers. In following, a number of additional reasons are mentioned to validate the attempts for accessing deep web data.

1.2.1 Current Search Engines Fail in Satisfying Some of Our Information Needs

Trying the queries like "what's the best fare from New York to London next Thursday" [2] or "count the percentage of used cars which use Gasoline Fuel" [12] on the search engines like Google, Yahoo, MSN, Bing and others would show that these search engines are missing the ability in providing users with answers to their information needs. For such a query, these general search engines might not even provide users with the best websites through which users can obtain their interesting data.

1.2.2 Huge Amount of High Quality Data

There are a number of surveys performed for estimating the size of deep web. In a survey performed in 2001, it was estimated that there are 43,000 to 96,000 deep websites with an informal estimate of 7,500 terabytes of data compared to 19 terabytes of data in the surface web [5, 9]. In another study, it was estimated that there were 236,000 to 377,000 deep websites having an increase rate of 3-7 times in the volume during 2000-2004 period [9, 6]. Ardian et al. [3] estimated that deep web contains more than 450,000 web databases which mostly contain structured data ("relational" records with attribute-value pairs). These structured data sources have a dominating ratio of 3.4 to 1, versus unstructured sources.

A significant portion of this huge amount of data is estimated to be stored as structured/relational data in web databases [7]. More than half of the deep web content resides in topic specific databases. This makes the search engines capable of providing highly relevant answers to every information need. This defines the high quality of the deep web data.

1.2.3 Benefits of Using Deep Web Data

Through deep web harvesting, mission-critical information (like information from competitors, their relationships, market share, R&D, pitfalls, and etc) existing in publicly available sources becomes available for companies to create business intelligence for their processes [5]. This access can also enable viewing content trends over time and monitoring deep websites to provide statistical tracking reports for the changes [5]. Estimating aggregates over a deep web repository like average document length, estimating repository sizes, generating content summaries and approximate query processing [12] are also possible with having deep web data at hand. Meta-search engines, price prediction, shopping website comparison, consumer behavior modeling, market penetration analysis, and social page evaluation are a number of example applications that accessing deep web data could enable [12].

1.3 The Goal

In this project, we aim at monitoring changes of entities in deep web sources. By entities, it is meant people, organizations, and every object that can be of a user interest. It is believed that knowing about the changes of these interesting entities over time in one or several deep web sources can reveal valuable information about those objects. It is also important to mention that only the publicly available deep web sources are targeted in this project. This is due to avoiding the legal limitations on accessing and providing data.

1.4 Structure of the Report

Giving an introduction over the topic of this research, definitions and motivations behind the research work in Section 1, the state-of-the-art in providing access to the data existing in deep web sources is discussed in Section 2. In Section 3, the challenges which should be addressed to have a thorough working system in accessing deep web data from finding the interesting deep web repositories to return results to user queries are listed and explained. In more details, the challenges chosen to be addressed through this research work are also discussed. In Section 4, the validation scenarios for these problems are presented.

2. STATE OF THE ART

In this section, a general overview is provided on the suggested approaches making data residing in deep web sources available to users.

2.1 Approach 1: Giving Indexing Permission

Data providers allow product search services to index the data available in their databases. This gives a complete access to the data in the databases. However, this approach is not applicable in a competitive and uncooperative environment. In an uncooperative environment, the owners of a deep website are reluctant to provide any information which could be used by their competitors. For instance, information about size, ranking, and indexing algorithms, and underlying database features are denied to be accessed.

2.2 Approach 2: Crawling all Data Available in Deep Web Repositories

This approach is based on the idea of extracting all the data available in deep web repositories which are of users' interests and give answers to their information needs by posing queries on this extracted data [11, 10]. This allows the web data sources to be searched and mined in a centralized manner.

In order to extract data from deep web data sources, their search forms are used as the entry points. Having filled in the input fields of these forms, the resulting pages are retrieved. After storage of this extracted data, it could be possible to answer user queries.

However, in order to realize this approach, a number of problems should be resolved first. The challenges like: smart form filling, structured data extraction, and having an automatic, scalable and efficient approach [8].

In "Crawling Deep Web Entity Pages" [7], two different types of deep websites are introduced; document-oriented and entity-oriented. In this research by Yeye et al. [7], document-oriented deep websites are defined as the websites which mostly contain unstructured text documents such as Wikipedia, Pubmed, and Twitter. On the other hand, entity-oriented deep websites are considered to contain structured entities; almost all shopping websites, movie sites, and job

listings [7]. Having compared these two types, entity-oriented deep websites are suggested to be very common and represent a significant portion of the deep websites. In this work, by focusing on entity-oriented deep websites, the entities are used for query generation, empty page filtering and URL deduplication.

2.3 Approach 3: Virtual Integration of Search Engines

In this method of providing data available in deep web repositories to users, extracting all the data from these sources is not targeted. Instead, it is tried to understand the forms provided by different deep data sources and provide a matching mechanism which enables having one mediated form. This mediated form sits on top of the other forms and is considered as the only entry point for the users. The queries submitted to this mediated form are translated into queries which are acceptable by other forms from the deep web repositories. In this process, techniques like query mapping, and schema matching are applied.

As the systems based on this approach need to understand the semantics of the provided entry points of deep web repositories, it could be of great effort and time to apply them in more than one or a number of related domains. The point that boundaries of domains on web data are not easily definable and also identifying which queries are related to each domain make the costs of building mediator forms and mappings high [8].

2.3.1 MetaQuerier

In "Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web" [4], a system based on the above mentioned approach is suggested. The suggested system abstracts the forms provided by web databases through providing a mediated schema [11].

2.3.2 IntegraWeb

In "On using high-level structured queries for integrating deep-web information sources" [10], a system named IntegraWeb is suggested. The main concept in this suggested system is issuing structured queries in high-level languages such as SQL, XQuery or SPARQL. This usage of high-level structured queries could lead to integrate the deep web data with less cost than using mediated schemes through abstracting away from web forms. In the virtual integration approaches, the unified search form abstracts away from the actual applications. Using structured queries over these mediated forms helps to have a higher level of abstraction [10].

2.4 Approach 4: Surfacing Approach

In "Google's Deep Web Crawl" by Madhavan et al. [8], the goal is to get enough appropriate samples from each interesting deep web repository so that the deep web could have its right place in the results returned by search engines for the entered queries. This task is performed by pre-computing the most relevant submissions for the HTML forms as the entry points to those deep web repositories. Then, the offline generated URLs from these submissions are indexed and added to the indexes of surface web. The rest of the work is performed as it is a page in surface web. The system represents search result and snippet to user which could redirect him to the underlying deep website [8]. This allows the user to access the fresh content.

3. CHALLENGES/PROBLEM STATEMENT

Before deciding which challenges to address, it is necessary to decide about the following issues:

1. Which deep web sources do we aim at; the information necessary to access the data behind web forms and the types of interfaces are examples of factors that could be taken into account in classification of deep web repositories. Therefore, it should be decided what types of deep web sources are targeted; (a) The ones which need information to login, and (b) The ones having interfaces of one of these types: one text input, several text and selection inputs, graph browsing (twitter, facebook) interface, or a hybrid of all these three interfaces [12].

2. What is the reason behind accessing the deep web data

   (a) If it is answering queries over a limited number of domains, then, it could be decided to have the mediated form approach.

   (b) If the reason is improving the position of answers resulted from the deep website in the returned results page from a search engine, then, a number of distinct samples which could cover all the different aspects of a deep website would be enough.

   (c) If the goal is to keep track of changes in the data provided by deep web sources, or to provide statistics over it, then, complete extraction and storage of data from deep web sources would be a necessary task to perform.

As mentioned before in the Introduction Section, in this project, we aim at monitoring changes in data available publicly to users. This makes challenges which should be addressed limited to crawling and extracting data from deep web data sources. These challenges are described in the following subsection.

3.1 Challenges for Implementing a Crawling Deep Web Mechanism

In this section, it is described that what obstacles should be resolved to have a query answering system which can find answers to queries from deep web sources. This system is considered to include all steps from finding interesting deep web repositories to pose queries and returning results. For implementing such a comprehensive approach, the challenges mentioned in the following subsections should be resolved.

3.1.1 Deep Web Source Discovery

In order to answer the information needs of a user, it is necessary to know from which data sources that information could be obtained. In the case of surface web, general search engines use the indexes and matching algorithms to locate those sources of interest. While in deep web sources, the data is blocked behind web search forms and far from search engines reach. Therefore, first of all, it should be discovered that which deep web data sources potentially contain the data necessary to answer a user query. To do so, the following questions should be answered. How to determine

the potential deep web sources for answering the query, considering the huge amount of websites available on the Web [12]? This could be done by narrowing down the search to find deep web sources of our interest while having all the interesting deep web sources covered. How to decide if a URL is a deep web repository [12]? To answer this question, it is necessary to be able to find forms in a website and decide if the form is the entrance to a deep web source or not. How to determine if the URL is of a given topic and related to the given queries [12]

3.1.2 Access Data Behind Web Forms

Knowing about the web forms of deep web repository which is of the user interest, it is the time to find the answers to his given query. As mentioned earlier in this section, there are a number of approaches which could be applied in accessing the data available in deep web data sources. Selecting the approach which aims at crawling all the data in a deep web website, the following challenges should be addressed.

- How to fill in the forms efficiently and automatically [12]? Which input fields of the form should be filled in? What are the bindings and correlations among the inputs? For example, bindings among input fields determine the inputs that should simultaneously be assigned with values to be able to get results. It also might refer to the inputs which could lead to more results if submitted with values at the same time. In addition, correlations in a form could refer to fields whose values depend on each other; such as minimum and maximum fields of an attribute in a form. What values to submit for those inputs so that the crawling is efficient; less queries but more results, less empty pages, less duplications of pages.
- How to extract data/information/entities from the returned results pages [12]? How to walk over all the returned results pages? How to extract data from each page? which data to extract? different websites have different scripting? How to detect empty pages and resolve the deduplication of pages and information? When to stop crawling? what algorithms to use? what features of deep web to measure? what is the size of deep web source? How to keep the cost of crawling low and the process efficient with implementations about the process itself such as following links found in the returned results pages? How to cover all the data available in the source considering the limitations on the number of query submissions and returned results?
- How to store the extracted data? How to perform entity identification, entity deduplication, and detecting relations among entities to have a high-quality information extraction? How to keep the quality of data extracted from structured deep web sources? As mentioned before in Section 1, deep web data is of a high quality as it is residing as structured data in domain specific databases.
- How could the entity identification, and detecting the relations among the entities help improving the crawling process as it goes on? How the results/data extracted from deep web source could be treated as feedback used in modifying and adjusting the crawling process to perform better in continuing the crawl.
- How to monitor entities/data changes over one/more deep web sources? For example, assuming an entity is found in several deep web sources, how the change in one website should be treated and interpreted and which version should be judged as being reliable?
- How to present the extracted data or monitored data to the users?

3.2 Challenges Targeted in This Research Work

Although answering all the challenges mentioned in the previous subsection is appealing, considering the limited time of a PhD Thesis work, only a small number of them could be covered. In the selection of challenges to be addressed in this research work, the capabilities and interests of users and also, the potentiality of contribution were the main issues considered. Therefore, the following list of questions are decided to be targeted during this PhD work.

- Q1: How to measure the size of a deep web source especially in a non-cooperative environment?

To the best of our knowledge, there is no crawling approach which claims the capability of extracting all the data existing in a deep web repository. The crawling approaches continue the process till they face the query submission limitations posed by search engines or consume all the allocated resources. To prevent this undesirable situation, a mechanism should be applied to stop the crawling wisely. This means to make a trade-off among the resources being consumed, limitations and the percentage of crawling coverage of a deep web source. To do so, knowing about the size of the targeted source is one of the most important factors. Therefore, it is set as the first question aimed to be resolved through this research work.

The process of extracting data from deep web sources is so costly that devising a more efficient crawling approach is one of the main topics of interest among researchers focusing on this area. As one of the solutions to this problem, in this work, it is aimed at applying approaches which benefit from the previously visited pages, or extracted information to make the remaining part of the process more efficient. This could be formulated as the second question of this research proposal.

- Q2: How could a deep web crawling approach be implemented with a feedback mechanism which could improve the crawling process as it goes on? For example, it should be studied that how the visited pages, extracted data, entity identification, or detecting the relations among those entities could be treated as feedback used in modifying and adjusting the crawling process to perform more efficiently and accurately in the remaining of the process.

Having an efficient deep web data crawler at hand, it is of our interest to be aware of the changes occurring over time in the data existing in deep web data repositories. This information could be helpful in providing the most up-to-date answers to information needs of users. Therefore, the third main question targeted in this research is dedicated to monitoring changes data in deep web sources;

- Q3: How to monitor changes of deep web data which could be unstructured or in the form of entities residing in one or more numbers of deep web sources?

To answer this question, the following challenges should be addressed:

- Q3-1: What is the most efficient way of detecting changes in a deep web data repository? Is there any more efficient way rather than crawling all the data residing in a deep web source in predefined intervals?
- Q3-2: What entities are more interesting to be monitored? For example, in a job vacancies website, one might be interested to follow a specific vacancy or a company's statistics about available or already filled vacancies. Difference in the entities and data monitored could lead to differences in the implementations of changes detection procedure.
- Q3-3: Assuming an entity is found in several deep web sources, how the change in one source should be treated and interpreted; which version should be judged as being reliable? Also, how such an environment could help in improving the change detection procedure?

4. VALIDATION SCENARIOS

The answers resulting from this research project for the crawling and monitoring questions mentioned in Section 3 will be validated through a case study in job vacancies domain. In this sense, the cooperation of WCC Company [1] would play an important role throughout this research project.

As mentioned in Section 3, the deep website crawling is effected by a number of different factors. For example, the type of input interfaces of the search engines and their access policies could effect the crawling process. The differences in ranking and indexing algorithms applied in search engines could also have influences on the crawling task. This makes it necessary to have access to websites available on the Web during the validation process. This could be provided through the mentioned cooperation of WCC. In case that the company could not provide us with such websites, the deep websites about which the information is publicly available will be targeted. For example, in the case of size estimation question, websites from which the size is known were used. Also, for a crawling task, websites like Wikipedia which provide access to all data residing in their databases could be helpful. These websites could be accesses through the Web or being already downloaded and used locally.

While this validation method would satisfy the needs for research question 1 and 2 mentioned in Section 3, it seems not sufficient for validating the solution for part of question 3. In research question 3, it is possible to aim at monitoring a single website or a number of deep websites. Considering monitoring one website having the similar requirements of a crawling task, the validation could be done in the same way as question 1 and 2. However, monitoring changes over a number of websites needs to have access to more than one website. The ideal situation to validate an approach for such a question is to have access to a set of deep websites running on the Web as they continue providing users with services they are designed for. While there are efforts to make this happening, especially through cooperation with WCC Company, as an alternative validation scenario, the locally performing websites are considered to be implemented.

9[1]http://www.wcc-group.com/

5. ACKNOWLEDGEMENT

This publication was supported by the Dutch national program COMMIT.

6. REFERENCES

[1] Deep web. `http://en.wikipedia.org/wiki/Deep_Web`, 2012.

[2] The New York Times Company Alex Wright. Exploring a deep web that google can not grasp. `http://www.nytimes.com/2009/02/23/technology/internet/23search.html?pagewanted=all`, 2012.

[3] Fajar Ardian and Sourav S. Bhowmick. Efficient maintenance of common keys in archives of continuous query results from deep websites. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 637–648, Washington, DC, USA, 2011. IEEE Computer Society.

[4] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the web, 2004.

[5] BrightPlanet Corporation. Deep web intelligence. `http://www.brightplanet.com`, 2012.

[6] Bin He, Mitesh Patel, Zhen Zhang, and Kevin Chen-Chuan Chang. Accessing the deep web. *Commun. ACM*, 50(5):94–101, May 2007.

[7] Yeye He, Dong Xin, Venky Ganti, Sriram Rajaraman, and Nirav Shah. Crawling deep web entity pages. Submitted to *VLDB 2012*.

[8] Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google's Deep Web crawl. *Proc. VLDB Endow.*, 1(2):1241–1252, August 2008.

[9] Umara Noor, Zahid Rashid, and Azhar Rauf. Article: A survey of automatic deep web classification techniques. *International Journal of Computer Applications*, 19(6):43–50, April 2011. Published by Foundation of Computer Science.

[10] Carlos R. Osuna, Rafael Z. Frantz, David Ruiz-Cortes, and Rafael Corchuelo. On using high-level structured queries for integrating deep-web information sources. In *International Conference on Software Engineering Research and Practice*, pages 630–636, 2011.

[11] Ping Wu, Ji-Rong Wen, Huan Liu, and Wei-Ying Ma. Query selection techniques for efficient crawling of structured web sources. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *ICDE*, page 47. IEEE Computer Society, 2006.

[12] Nan Zhang and Gautam Das. Exploration of deep web repositories. *PVLDB*, 4(12):1506–1507, 2011.