SmartCrawler: for Efficiently Gathering Deep Web Contents

Shivkanya Mane¹, Rajashree Galave², Sonal Lohabade³

¹²³Student, Department of Computer Engineering, Sinhgad Institute of Technology Lonavala, Maharashtra, India

ABSTRACT

Deep web is concerned with the hidden contents behind search interfaces on the World Wide Web. Due to the amount of web information is increasing speedily, vibrant nature of deep web and also because of the massive volume of deep web resources gathering information from a web record, achieving high frequency and wide coverage becomes more challenging for ordinary search engines and general purpose crawlers. So, we described the new approach to a resource discovery system called a SmartCrawler, for efficiently and effectually gathering information from deep web, by avoiding visiting a huge number of unnecessary pages. We proposed a two-stage framework which is having two stages- website locating and in-site searching is for discovering more contents which are relevant to a given search matter. Within the website locating stage SmartCrawler performs website based searching for center pages. In this stage websites are ranked by SmartCrawler. This ranking mechanism prioritizes websites for a given search matter. Within the in-site searching stage Adaptive Link Ranking is used for speedy searching within the websites for discovering more contents which relates to a given search matter. For achieving wider coverage websites a link tree data structure is designed.

Keywords: Adaptive learning, Deep web, Feature selection, Ranking, Two stage crawler.

1. INTRODUCTION

Basically, a web crawler is the agenda that visits web pages, gathers and after that categorizes information on the web. The web crawler contains three parts: crawler, index and software. First is the crawler also called as spider that visits the pages, fetches the information and after that it follows links in different pages within a website. The crawler comes back to a crawled website over regular interruption of time. The information found in the first step is given to the second step, the index which is also called as a record. The index is like a database which is updated with newly found contents in the indexed database. Third part is software, which is a program that shifts billions of pages which are already recorded in the indexed database to determine matches to a required search matter and level them in the order as what it finds as more relevant.

Deep web [1],[2],[12] also called as invisible web or buried web. As deep web is something you cannot find with a single search, it is complex to locate deep web interfaces, because they are not registered by search engines. They are usually rarely distributed and vibrant in nature.

To defeat the problems with deep web interfaces in prior work related with deep web crawling techniques and tools, we proposed a SmartCrawler which is having a two-stage framework. SmartCrawler is a Focused Crawler [3],[4],[5],[6],[7]. The Focused Crawler focus on a specific subject matter and it fetches only the most relevant searchable forms to a given subject matter. Thus SmartCrawler, performs a superior level of data examination and data mining from the deep web. SmartCrawler has two stages: website locating and in-site searching. Within the first stage, SmartCrawler performs website based searching for center pages [1] with the help of search engine by avoiding visiting huge number of unnecessary pages. To achieve more precise, proficient and useful results SmartCrawler ranks website by using ranking mechanism [1] to prioritize extremely relevant for a given subject matter. If the amount of websites fetched in frontier is lower than the user defined threshold then the SmartCrawler performs Reverse Searching [1] to discover more searchable web forms. Within the second stage, SmartCrawler performs fast in-site searching and incremental two-level website ranking for finding closely relevant websites and to discover more data sources. During in-site searching a link-tree data structure is designed for achieving extensive coverage. Adaptive Learning Algorithm [1] performs Online Feature Selection and automatic construction of Link Rankers.

2. RELATED WORK
In the prior work, methodically it is presented that the crawling deep web has three steps: locating deep web content sources, selecting relevant sources, dig out essential contents [11].

A. Locating Deep Web Content Sources

In the previous work, the MetaQuerier [3], the Database Crawler is intended for involuntarily discovering query interfaces which primarily finds source pages by an IP based sampling, then performs shallow crawling to crawled pages contained by a web server starting from a given source page. The disadvantage is, the IP based sampling take no notice of the fact that one IP address may have more than a few virtual hosts [4], accordingly missing many websites. To defeat the disadvantage of IP based sampling in the Database Crawler, a stratified random sampling of hosts is used to describe national deep web [5], by means of the Host-graph provided by the Russian search engine named Yandex.  I-crawler [6] merged pre-query and post-query approaches for sorting the searchable forms.

B. Selecting relevant sources

Previously the Best-First Focused Crawler is designed which uses a page classifier [10]. Experimentally the Best-First Focused Crawler gathered only 94 movie search forms after crawling about 10,000 movie related pages [9]. An extended version of Best-First Crawler is proposed in [13], where instead of following all the links in relevant websites, the crawler used the supplementary classifier, the apprentice to select the majority promising links in a relevant website. The baseline classifier gives choice as feedback so that the apprentice can learn the features of good links in the frontier. The prior work also proposed the two types of crawlers- Generic Crawler and Focused Crawler [3],[4],[5],[6],[7]. Generic Crawlers do not focus on a particular topic and it fetches all searchable forms while Focused Crawlers focus on a particular topic and fetches only relevant searchable forms. The Form Focused Crawler (FFC) [9] and Adaptive Crawler for Hidden web Entries (ACHE) [8] are the Focused Crawlers. The FFC[9] and ACHE [8] aims to proficient and automatic detection of other forms in the given search matter. The experimentation results has carried out for record contents, it has been proved that the correctness of FFC is only about 16 percent [8] [9] which is small in terms of mining relevant contents.

Diverse from the techniques and tools mentioned above we proposed SmartCrawler, a Focused Crawler for gathering deep web content sources. SmartCrawler targets at deep web interfaces and employs a two-stage design, which is not only classifies websites in the first stage to filter out irrelevant websites but also categorizes searchable forms in the second stage. Instead of simply classifying links as relevant or not, SmartCrawler first ranks websites and then prioritizes links within a site with another ranker.

3. SYSTEM ARCHITECTURE

The system architecture of proposed work is as shown in Fig.1:

![System Architecture Diagram]

C. Website Locating

This stage consists of: website locating, website ranking and website classification. The General Crawler finds only recently found links but SmartCrawler not only minimizes the number of visited URLs, but also maximizes the number of deep searches.
D. Reverse Searching

If the website contains searchable forms, it is relevant and if the amount of websites fetched in frontier is larger than the user-defined threshold then the page has relevancy. Reverse search is place when

- The crawler bootstraps,
- The size of frontier is lower than the user-defined threshold.

E. Incremental Website Prioritizing

For creation of crawling system presumable and succeed wider coverage on websites the incremental website prioritizing is deliberated. First, the information obtained by precedent crawling like deep websites is employed to initialize Website Ranker and Link Ranker. Then, unvisited websites are assigned to website frontier and are prioritized by the Website Ranker, and visited websites are added to fetched website list.

F. Website Classifier

After ranking, the Website Classifier classifies the websites either as relevant for a Focused Crawler likewise FFC and ACHE.

4. RANKING MECHANISM

Two ranking mechanisms: Website Ranking and Link Ranking & also the related mathematical formulations are as described below:

G. Website Ranking

SmartCrawler ranks websites to prioritize relevant deep websites for a given subject matter. At this end, website similarity and website frequency these two features are measured for ranking. Website similarity measures the relevancy between a new website and a known deep websites. Website frequency measures the number of times a website appears in other websites. The rank of a new site is a function of website similarity and website frequency.

Given the homepage URL of a new website s:

\[ s = \{ U_s, A_s, T_s \} \]  

(1)

The mathematical formulation for website similarity Feature Space of known Sites (FSS) is as follows:

\[ ST(s) = Sim(U_s, U_s) + Sim(A_s, A_s) + Sim(T_s, T_s) \]  

(2)

Where, \( Sim( ) \) scores similarity of the associated features between s and known deep websites.

The function \( Sim( ) \) can be calculated as the cosine between vector V1 and V2 as follows:

\[ sim(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| \times |V_2|} \]  

(3)

The mathematical formulation for measuring website frequency is as follows:

\[ SF(s) = \sum_{known\, websites\, list} I_i \]  

(4)

Where, \( I_i = 1 \), if \( s \) is appeared in known deep websites, else \( I_i = 0 \).

For the rank we use a linear combination of website similarity and website frequency as follows:

\[ Rank(s) = \alpha \times ST(s) + (1 - \alpha) \times \log(1 + SF(s)) \]

where \( 0 \leq \alpha \leq 1 \)

H. Link Ranking
Link prioritizing is based on the Feature Space of Links (FSL) with searchable forms. Here the link similarity is measured for prioritizing links, similar as website similarity.

Given a new link:

\[ l = \{P_j, A_j, T_j\} \]  

(6)

The mathematical formulation for link similarity to FSL is as follows:

\[ LT(l) = Sim(P, P_j) + Sim(A, A_j) + Sim(T, T_j) \]  

(7)

5. CONCLUSION

In this paper, we proposed a two-stage SmartCrawler for efficient gathering of deep web contents. We tend to prove experimentally that our SmartCrawler can do efficient and effectual gathering of the deep web content resources and also achieves extensive coverage. In future work, we would like to extend our work by merging the pre-query and post-query approaches for deep web.

6. ACKNOWLEDGMENT

We are thankful to our project guide Prof. N. K. Patil Sir. We would also like to give our sincere gratitude to the all staff of Computer Engineering Department for coordination and all the friends.

REFERENCES


