International journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.2., Issue.3., 2014

RESEARCH ARTICLE

ISSN: 2321-7758

# AN ONTOLOGY BASED SEMANTIC FORUM IDENTIFIER

**REZNI.S**
PG Scholar, Department of CSE, Hindusthan College of Engineering and Technology,
Coimbatore, Tamilnadu, India

Article Received: 24/05/2014 | Article Revised on: 30/05/2014 | Article Accepted on:02/06/2014

**REZNI.S**

ABSTRACT
Web Crawlers are the heart of the search engines. Crawling is the initial and the most important step during the web search procedure. So while crawling, many problems will occur and the solutions to those crawling problems are presented. As the crawler visits these urls, it identifies all the hyperlinks from this urls and adds them to the list to visit. It provides a vocabulary including unambiguous definitions for terms that can serve as a support for communication between software agents and for users, and it define the semantics for different domains for the purpose of interactions on the web, and help in the creation of knowledge base that will allow people to work on a particular domain. As a result, the web crawlers support multiple keywords and also provide an accurate and effective result to the users.

***Keywords – Search engine, Crawler, Semantic web, Ontology, Machine Learning***

## INTRODUCTION

The crawler extracts URLs appearing in the retrieved pages, and gives this information to crawler for control the module. This module determines what links to visit next, and feeds the links to visit back to the crawlers. The crawler also passes the retrieved pages into a page repository. The learning part first learns ITF regexes of a given forum from automatically constructed URL training samples. The online crawling part then applies learnt ITF regexes to crawl all threads efficiently. Given any page of a Forum, FoCUS, a supervised web-scale forum crawler, first finds its entry URL using the *Entry URL Discovery* module. Then, it uses The *Index/Thread URL Detection* module to detect index URLs and thread URLs on the entry page; the detected index URLs and thread URLs Are saved to the URL training sets. Next, the destination pages of the detected index URLs are fed into this module again to detect more indexes and thread URLs until no more indexes URL is detected. After that, the *Page-Flipping URL Detection* module tries to find page-flipping URLs from both index pages and thread pages and saves them to the training sets. Finally, the *ITF Regexes Learning* module learns a set of ITF regexes from the URL training sets. Once the learning is finished, FoCUS performs online crawling as follows: starting from the entry URL, FoCUS follows all URLs matched with any learnt ITF regex. FoCUS continues to crawl until no page could be retrieved or other condition is satisfied. Crawlers continue visiting the web, until local resources, such as storage, are fatigued. Our contribution of work follows as:

1. Identify the base URL in the website.
2. Identify type of protocol used for any web page.
3. Retrieve the web pages, we apply pattern recognition over text and pattern symbolizes check text only.
4. Check how much text is available on web page.

We proposed and implemented FoCUS, a supervised forum crawler. We reduced the forum crawling problem to a URL type recognition problem and showed how to leverage implicit navigation paths of forums, i.e. EIT path, and designed methods to learn ITF regexes explicitly. We also showed that FoCUS can effectively apply learnt forum crawling knowledge on 160 unseen forums to automatically collect index URL, thread URL, and page-flipping URL training sets and learn
ITF regexes from the training sets. Experimental results on 160 forum sites each powered by a different forum software package confirm that FoCUS can effectively learn knowledge of EIT path from as few as 5 annotated forums.

**RELATED WORK**

***Intelligent Crawling Of Web Applications:***

The steady growth of the World Wide Web raises challenges regarding the preservation of meaningful Web data. Tools used currently by Web archivists blindly crawl and store Web pages found while crawling, disregarding the kind of Web site currently accessed which leads to suboptimal crawling strategies and whatever structured content is contained in Web pages which results in page-level archives whose content is hard to exploit.

A Web application is any application that uses Web standards such as HTML and HTTP to publish information on the Web, accessible by Web browsers. Examples include Web forums, social networks, relocation services, etc. the claim that the best strategy to crawl these applications is to make the Web crawler aware of the kind of application currently processed, allowing it to refine the list of URLs to process, and to annotate the archive with information about the structure of crawled content. We add adaptive characteristics to an archival Web crawler: being able to identify when a Web page belongs to a given Web application and applying the appropriate crawling and content extraction methodology.

***Crawler for Dark Web Forums:*** The unprecedented growth of the Internet has propagated the escalation of the Dark Web, the problematic facet of the web associated with cybercrime, hate, and extremism. Despite the need for tools to collect and analyze Dark Web forums, the covert nature of this part of the Internet makes traditional web crawling techniques insufficient for capturing such content. In this study we propose a novel crawling system designed to collect Dark Web forum content. The system uses a human assisted accessibility approach to gain access to Dark Web forums.

Several URL ordering features and technique enable efficient extraction of forum postings. The system also includes an incremental crawler coupled with a recall improvement mechanism intended to facilitate enhanced retrieval and updating of collected content. Experiments conducted to evaluate the effectiveness of the human-assisted accessibility approach and the recall improvement based incremental update procedure yielded favourable results. The human assisted approach significantly improved access to Dark Web forums while the incremental crawler with recall improvement also outperformed standard periodic and incremental update approaches.

***Tunnelling, And Digital Libraries:*** Crawling the Web to build collections of documents related to pre-specified topics became an active area of research during the late 1990's, crawler technology having been developed for

use by search engines. Now, Web crawling is being seriously considered as an important strategy for building large scale digital libraries. This concept covers some of the crawl technologies that might be exploited for collection building. For example, to make such collection-building crawls more effective, focused crawling was developed, in which the goal was to make a "best-first" crawl of the Web.

They are using powerful crawler software to implement a focused crawl but use tunneling to overcome some of the limitations of a pure best-first approach. Tunneling has been described by others as not only prioritizing links from pages according to the page's relevance score, but also estimating the value of each link and prioritizing them as well. Results indicate that a combination of focused crawling and tunneling could be an effective tool for building digital libraries.

**WEB CRAWLING**

Many Web services operate their own Web crawlers to discover data of interest, despite the fact that large-scale, timely crawling is complex, operationally intensive, and expensive. The extensible crawler, a service that crawls the Web on behalf of its many client applications. Clients inject filters into the extensible crawler; the crawler evaluates all received filters against each Web page, notifying clients of matches. As a result, the act of crawling the Web is decoupled from determining whether a page is of interest, shielding client applications from the burden of crawling the Web themselves.

Focus on the challenges and trade-offs in the system, such as the design of a filter language that is simultaneously expressive and efficient to execute, the use of filter indexing to cheaply match a page against millions of filters, and the use of document and filter partitioning to scale our prototype implementation to high document throughput and large numbers of filters. In most popular search engines, it is well-known that the ranks of websites for general queries are directly relevant to their economic benefits. Thus, a common phenomenon has emerged that many spam websites play tricks on search engine crawlers by artificially increasing links, in order to increase their weight in the Page Ranks algorithm.

***System Architecture***

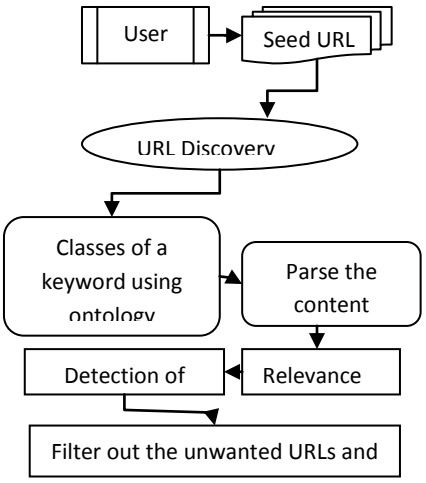


*Figure1 System Architecture*

A conventional Web crawler is used to fetch a high rate stream of documents from the Web. Depending on the needs of the extensible crawler's applications, this crawl can be broad, focused, or both. For example, to provide applications with real-time information across pods for processing. A pod is a set of nodes that, in aggregate, contains all filters known to the system. Because documents are partitioned across pods, each document needs to be processed by a single pod; by increasing the number of pods within the system, the

overall throughput of the system increases. Document set partitioning therefore facilitates the scaling up of the system's document processing rate.

**URL Discovery:** FoCUS learns ITF regexes that can be used in online crawling. However, an entry URL needs to be specified to start the crawling process. To the best of our knowledge, all previous methods assumed that a forum entry URL is given. In practice, especially in web-scale crawling, manual forum entry URL annotation is not practical. Forum entry URL discovery is not a trivial task since entry URLs vary from forums to forums. To demonstrate this, we developed a heuristic rule to find entry URL as a baseline. The heuristic baseline tries to find the following keywords ending with "/" in a URL: forum, board, community, and discus. If a keyword is found, the path from the URL host to this keyword is extracted as its entry URL; if not, the URL host is extracted as its entry URL.

**Index/Thread URL Detection:** It uses the Index/Thread URL Detection module to detect index URLs and thread URLs on the entry page; the detected index. A majority voting method is adopted to determine the URL type since classification results on individual destination page might be erroneous. By utilizing aggregated classification results, FoCUS does not need very strong page classifiers. In summary, we create index and thread URL training sets and collects all URL groups and computes their total anchor text length. Then selects the URL group with longest anchor text length as the index/thread URL group. If the pages are not index or thread page, the URL group is discarded.

**Flipping URL:** Page-flipping URLs point to index pages or thread pages but they are very different from index URLs or thread URLs. Proposed "connectivity" metric to distinguish page-flipping URLs from other loop-back URLs. However, the metric only works well on the "grouped" page-flipping URLs, more than one page-flipping URL in one page. But in many forums, there is only one page-flipping URL in one page, which we called single page-flipping URL.

**PROPOSED SCHEME**

**Multi keyword Web Crawling using Ontology:** Ontologies provide a formal semantics that can be employed to process and integrate information on the web. Gruber describes ontology as an explicit specification of conceptualization. Ontologies play a pivotal role in providing a vocabulary comprising unambiguous definitions for terms that can essentially serve as a formal support for communication between software agents. They provide a communication mechanism for users and software agents, clearly define the semantics for different domains for the purpose of interactions on the web, and help in creating a knowledge base that will enable people to work on a particular domain .Web Ontology Language(OWL), recommendation from W3C, is widely used to construct a domain ontology.

***Crawling steps:***

Step 1: Get the seed URL from initURL.txt file and class keyword from OWL file.

Step 2: If the webpage is valid that is it of the defined type like (html, php, jsp etc.) then it is added to queue.

Step 3: Parse the content with the help of Jena Class Extractor with the help of html parser for all the key word found from the Ontology classes.

Step 4: Get the response from server if it is ok add the webpage to index and caches file to a folder. With the help of cache and index searching can be done and ranking the page according to the keyword search.

Step 5: After crawling the current page extract the new URL's from the page and store the URL in a ontology based database with the page rank.

**Proposed algorithm**

1: Retrieve all classes of a keyword using ontology.

2: Maintain a variable AverageRank (initially = 1).

3: pick up a URL from URL_Buffer and retrieve corresponding web page.

4: Extract all URLs (Extracted_URLs) from webpage

5: Calculate the rank for each class(found using ontology in step 1).
6: Calculate TotalRank = Sum(rank of all classes).
7: if (AverageRank<=TotalRank) then add Extracted_URLs to URL_Buffer.
else (Discard all Extracted_URLs).
8:AverageRank=(AverageRank+TotalRank) /Total URL_Crawled.
9: Go to step (3)

**Implementation**

In a hybrid approach for searching the Semantic Web is described: classical search techniques are combined with spread activation techniques applied to a semantic model of a given domain. The figure is a high level representation of the context and applications of our crawler system. We can distinguish four important parts:

(i) The input of the crawler,
(ii) The ontology commitment by the user used to guide the crawler,
(iii) The processing and storing of the processed input
(iv) A selection of the possible text of that processed data.

Initially, user inputs a seed URL to the web crawler. Crawler then fetches the corresponding web page from the Internet. This page is then stored in a buffer and its contents are parsed. We separate the links and content of the fetched web page. Now, the extracted links are stored in the database. Also, if we want to further traverse these links we can click the desired link and this link is given as input to the crawler and same procedure is repeated. Additionally, we can search a particular keyword in the content of the fetched urls stored in database and check if it is present or not. If yes, the crawler prints the number of times the particular keyword occurs in the Web Page.

**Evaluations:**

In this we discussed about our proposed scheme and how to implement it. In this section, we illustrate all the method in separate module with detailed description such as synopsis of anticipated scheme, ITF Regexes Learning, Online Crawling and Entry URL Discovery.

The learning part first learns ITF regexes of a given forum from automatically constructed URL training examples. The online crawling part then applies learned ITF regexes to crawl all threads efficiently.

**Page Type:** In this module, we classified the forum pages into following page types.

**Entry Page:** The homepage of a forum, which contains a list of boards and is also the lowest familiar ancestor of all threads.

**Index Page:** A page of a board in a forum, which typically contains a table-like structure and which contains information of a board or a thread. The list-of board page, list-of-board and the thread page, and the board page are all index pages.

**Thread Page:** A page of a thread in a forum that contains a list of posts with user generated content belonging to the comparable discussion.

**URL Type:** In this module, we discuss about types of URL

**Index URL:** A URL that is on an entry page or index page and points to an index page. Its anchor text shows the title of its destination board.

**Thread URL:** A URL that is on an index page and points to the thread page. Its anchor text is title of the destination thread.

**Page Flipping URL:** Links connecting multiple pages of a board and multiple pages of thread. The URL that leads users to another page of a same board or same thread.

***ITF Regexes Learning***

In this section, we learn about ITF regexes, FoCUS which adopts two-step supervised training procedure. The first step is training sets construction. The second step is regexes learning.

**CONCLUSION**

In this paper, we briefly review the existing keyword Matching Crawler and Multi-Threaded Ontology-based focused crawlers. A comparison is made among these crawlers from the perspective of keyword matching and domain specific crawling, working environment, special functions, technologies utilized, evaluation metrics and evaluation results, in order to survey its current research status. The conclusion with respect to this comparison is made in this section. For making Crawler more intelligent we can use the NLP which is a more effective concept for providing the real world meaning to the ontology based web crawler for the key word matching and searching that can be implemented in the further scope of this paper.

**REFERENCES**

[1]. Wikipedia:http://en.wikipedia.org/wiki/ Ontology_(information_science)
[2]. T.R.G ruber,—What is an Ontology?,||http://wwwksl.stanford.edu/kst/what-is-an ontology.html
[3]. http://infolab.stanford.edu/ ~backrub/google.html
[4]. M.Peshave, A.Dezhgosha,How Search Engines Work and Web Crawler Application.
[5]. M. Ehrig, A. Maedche, OntologyFocused Crawling of Web Documents (2003)
[6]. B.Lee, Godel, and Turing, Thinking on the Web, Wiley, pp xv, pp xxv, xxvi, pp-108
[7]. Passin, T.B. Explorer's guide to the semantic web||,Manning, pp-18
[8]. T.R. Gruber, A translation approach to portable ontology specifications||, Knowledge Acquisition, 5:199–220 (1993)
[9]. "The Sitemap Protocol," http://sitemaps.org/protocol.php, 2012.
[10]. "The Web Robots Pages," http://www.robotstxt.org/, 2012.
[11]. WeblogMatrix,"http://www.weblogmatrix.org/ , 2012.
[12]. S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine." Computer Networks and ISDN Systems, vol. 30,nos. 1-7, pp. 107-117, 1998.
[13]. R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang, "iRobot: An Intelligent Crawler for Web Forums," Proc. 17th Int'l Conf. WorldWide Web, pp. 447-456, 2008.
[14]. A. Dasgupta, R. Kumar, and A. Sasturkar, "De-Duping URLs via Rewrite Rules," Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 186-194, 2008.
[15]. C. Gao, L. Wang, C.-Y. Lin, and Y.-I. Song, "Finding Question-Answer Pairs from Online Forums," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 467-474, 2008.
[16]. N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, "Deriving Marketing Intelligence from Online Discussion,"Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 419-428, 2005.
[17]. Wu Chensheng, Hou Wei, Shi Yanqin, Liu Tong —A Web Search Contextual Crawler Using Ontology Relation Mining.
[18]. Pooja gupta Mrs. Kalpana Johari,||Implementaion of Web Crawler ,Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09
[19]. Bidoki, Yazdani et el, —FICA: A fast intelligent crawling algorithm||, Web Intelligence, IEEE/ACM/WIC International conference on Intelligent agent technology, Pages 635-641, 2007.
[20]. Junghoo Cho, Hector Garcia-Molina, Lawrence Page, |Efficient crawling through URL ordering||, 7th International WWW Conference , April 14-18, Brisbane, 1998.
[21]. Peisu, Ke et el, A Framework of deep web crawler,
[22]. Wenqing Yuan,—Research on Prototype Framework of a Multi-Threading Web Crawler for E-Commerce, International Conference on MASS ,20-22 Sept. 2009.