Review of web crawlers

S.R. Sreeja* and Sangita Chaudhari

Department of Computer Science,
A.C. Patil College of Engineering,
Sector 4, Kharghar, Navi Mumbai,
Maharashtra, 410210, India
E-mail: sreejasr09@gmail.com
E-mail: sschaudhari@acpce.ac.in
*Corresponding author

Abstract: The web is a repository of large amount of data. Information available in the web is organised in the form of pages. Due to the presence of unlimited amount of information, searching and finding out appropriate information from the web is a task which needs expertise. Web crawlers are programmes that assist search engines by automating the task of visiting web pages and downloading their contents. They also help in ranking the downloaded web pages. Thus, the search engines can produce a list of web pages ordered by their relevance and can display this list as a result of the search. Crawling also helps to validate web pages, analyse them, notify about page-updation, visualise web pages and sometimes for collecting e-mail addresses for spam purposes. They can be of different types, each one using different strategies and techniques to crawl web pages. This paper presents a review of various types of web crawlers.

Keywords: deep web crawler; focused crawler; web forum; forum crawler; web intelligence; web crawler review.

Reference to this paper should be made as follows: Sreeja, S.R. and Chaudhari, S. (2014) 'Review of web crawlers', *Int. J. Knowledge and Web Intelligence*, Vol. 5, No. 1, pp.49–61.

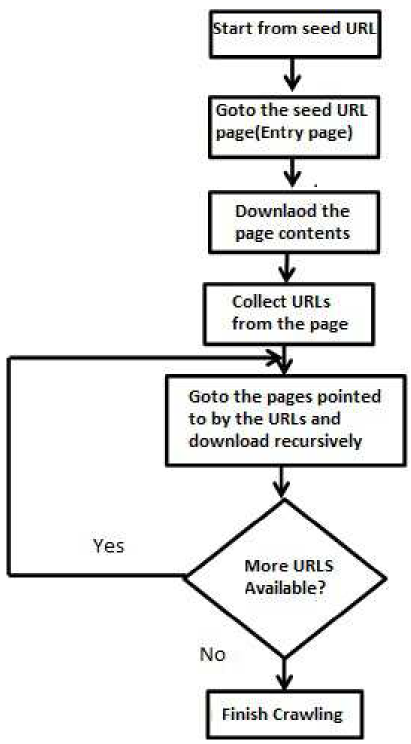
Biographical notes: S.R. Sreeja received her BTech in Computer Science and Engineering from Mahatma Gandhi University Kottayam. She is a PG student in Computer Engineering Department in A.C. Patil College of Engineering, Mumbai University. Her research interests include the concepts and issues related to web data mining and web crawlers.

Sangita Chaudhari received her ME in Computer Engineering from Mumbai University, India. Currently, she is an Assistant Professor at A.C. Patil College of Engineering, Kharghar, Navi Mumbai, India. Her research interests include digital image processing, advanced databases, information systems, and information security techniques. She has published more than 15 papers in national/international conferences and journals.

1 Introduction

Web crawler is a programme or a suit of programmes that is used to retrieve contents of web pages. This content retrieval is done mainly for the purpose of ranking the web pages. Some sites and most of the search engines use a web crawler for the purpose of updating their contents (Pinkerton, 1994). The major types of web crawlers are generic deep web crawlers, focused crawlers and forum crawlers. Many crawlers have been developed which belong to these three categories. All these types of crawlers will start crawling from a single uniform resource locator (URL) called the 'seed URL'. A deep web crawler crawls web pages without considering the relationship among the pages. They start from a 'seed URL' and crawls web pages recursively until there are no more links to follow. The basic steps followed by a web crawler are shown in Figure 1. Breadth first strategy (BFS) is commonly used by a deep web crawler for crawling. Focused crawler is a type of web crawler that crawls web pages which are specific to a pre-defined topic or domain. In comparison to a generic deep web crawler, focused crawlers consume less amount of system resources. Focused crawlers use a variety of techniques for domain-specific retrieval. Some of them use semantic-based methods while others use various properties of the web pages like number of links between them and similarity of their contents. Web forums also called internet forums provide a space for users to share knowledge (Internet forum, http://en.wikipedia.org/wiki/Internet_forums). Users can post their queries in a web forum. They will get the replies of their queries as well as many other useful information related to the queries from the forums. Forum sites are created by using forum softwares which can be generic or customised. Crawling web forum sites require specific forum crawling softwares. Forum crawlers themselves use different methods for performing the crawling.

Figure 1 Working of a basic web crawler



This paper concentrates on a survey of different types of web crawlers. There are two major issues that a web crawler has to deal with. A web page will contain so many links that are duplicate and leading to uninformative pages which a web crawler has to identify and ignore. Moreover, there can be 'spider traps' which may result in indefinite crawling and unlimited consumption of system resources. Web crawlers use different techniques to deal with these issues. The aim of doing this review is to help researchers to understand about various types of web crawlers and to identify which type of web crawler will be suitable for a particular purpose. Another objective is to explore the research possibilities in the area of web forum crawling. In the coming section, the strategies used by each type of web crawler and how each crawler was evaluated by the designers are explained in detail. Each crawler was evaluated by using some parameters. Some of the major parameters used for evaluating web crawlers were *crawling time*, *precision* and *recall* (Information retrieval parameters, http://http://en.wikipedia.org/wiki/Precision_and _recall). The evaluation criteria section explains these parameters. The assessment of web crawlers section contains a table that summarises the findings. It is followed by a research gap section which discusses about the areas where further researches are required. This section is followed by a conclusion. In the concluding part, the result of this comparative study is included.

2 Existing web crawlers

Existing web crawlers crawl web pages using different methods. Some of them visit and download all the web pages by following the hyperlinks and others crawl only limited number of pages based on some predefined criteria. They also exhibit differences in the strategies that they are using to select the hyperlinks. Some of the web crawlers are the following.

2.1 PyBOT

PyBOT can be categorised as a generic deep web crawler. PyBOT starts by taking the 'seed URL' and from that URL, it gets all other URLs. This is done by scanning the web page pointed to by that URL. Using the collected URLs, it crawls again until a point where no new URLs are found. It downloads all the web pages that it crawls. The output of PyBOT contains both the downloaded pages and the web structure in Excel CSV format. This can later be used for ranking/indexing. PyBOT uses traditional breadth first crawling (TBFC) strategy and is implemented using a tree search and a first in first out (FIFO) queue. TBFC uses a strategy in which the root node is crawled first. After that its successors are crawled. Then their successors are crawled and so on. This continues until all the nodes are crawled at a given depth in the search tree. Here, the root node is the starting URL and all the URLs in the page pointed by that URL are visited and downloaded and this continues. All the newly generated links are put at the end of the queue, so that shallow pages are expanded before deep pages. PyBOT has the advantage that this can be used in all kinds of cases. Also, it is very efficient when used in crawling normal websites (Najork and Wiener, 2001). But it has several drawbacks as it uses TBFC strategy for crawling. TBFC has a limitation that after a large number of web pages are fetched, it will start losing its focus which will result in the introduction of a lot of noise into the final collection. It may crawl many redundant and duplicate pages and may miss many useful pages. Download of large amount of useless pages wastes network bandwidth and negatively affects repository quality. It crawls the pages without understanding the correlation among them and so it cannot be used to crawl web forums (Leng et al., 2011). PyBOT was tested in the Curtin website. PyBOT visited and downloaded the pages and along with that it retrieved all the hyperlinks and saved them in Excel comma separated values (CSV) format. Web pages were saved as text files.

2.2 Focused crawler

A focused crawler is designed to crawl web pages relevant to a specific pre-defined topic. A focused crawler developed for collecting the financial data of a specific country works as follows:

- First determine the country for which the system has to be developed. The domain name extension of each country will be unique. For Bangladesh, the domain name ends with '.bd', and for Canada '.ca', etc.
- IP address is found out from the domain name and then the country name is determined by geo-mapping it.
- If the website contains any finance related key word, then it is categorised as a financial website of any specific country. Some specific keywords are used to understand the country and financial information.

This crawler uses a multi-threaded system for bandwidth management. Download speed and total bandwidth are calculated during crawling. If the bandwidth is high then the number of threads is increased; otherwise, it is decreased. Compared to the generic deep web crawler, focused crawlers yield good recall and precision by restricting themselves to a limited domain. This crawler has the advantage that it reduces the network bandwidth by downloading pages that are only relevant to the topic. Also, it puts efforts to manage the bandwidth by managing a multi-threaded system. This crawler has a limitation that it is specific to a particular application. Another limitation is that increasing and decreasing threads for bandwidth management is a tedious task (Dey et al., 2010).

2.3 Ontology-based semantic focused crawler

A semantic focused crawler is a focused crawler that makes use of semantic web technologies for performing the crawling (Shah et al., 2002). An ontology-based semantic focused crawler links web documents with related ontology concepts for the purpose of categorising them. It makes use of ontologies to analyse the semantic similarity between URLs of web pages and topics (Yuvarani et al., 2006). Topics are stored in the form of ontologies in an ontology base. When a query is provided, the compatible ontology will be retrieved from the ontology base. The query will then be sent to the commercial search engines, to retrieve the relevant URLs. After that, a multithreaded crawler will fetch web pages based on these URLs. After crawling the web pages, all URLs and their surrounding texts will be extracted from the web pages. These texts will then be matched with compatible ontology concepts to determine the relevance of the URLs to the query. The limitation of this type of crawler is that most of these crawlers fetch the surrounding texts of URLs as the descriptive texts of the URLs and compute the similarity between the URLs and ontology concepts based on these texts. But, the surrounding texts are not sufficient to describe the URLs correctly. If they are used, it may increase the fault rate of the similarity computing (Maedche et al., 2002).

2.4 Metadata abstraction-based semantic focused crawler

A metadata abstraction-based semantic focused crawler is a focused crawler that extracts meaningful information or metadata from relevant web pages and annotates the metadata with ontology markup languages. Such a crawler generally consists of a metadata creation component which will extract meaningful information from the downloaded web pages and will convert it into metadata. It consists of a data analysis component that generates analytical reports such as classification of web pages. Most commonly used algorithms for document classification are Naive Bayes and multiclass support vector machines (MSVMs). One of the drawbacks of metadata abstraction focused crawlers is that these crawlers mostly make use of the supervised classification models such as MSVM, SVM, etc. for web document classification. Many of these supervised classification models use predefined classifiers based on plain texts without enough semantic support. This will decrease the performance of document classification and the classified web documents will not fulfil users' requirements for the domain knowledge-based search. Another drawback is that a metadata-based focused crawler is specific to the application for which it is developed (Ding et al., 2004).

2.5 Combining ontology-based and metadata-based crawlers

This crawler combines the features of ontology-based focused crawlers and metadata abstraction focused crawlers. This type of crawler consists of many modules like a webpage fetcher for downloading the web pages, a policy centre to control the behaviour of the webpage fetcher by setting up several policies, a webpage parser to extract meaningful information from the web documents, a metadata generator to generate metadata, a metadata classifier to classify the generated metadata, and a webpage pool to act as a repository. Along with these modules, a knowledge base also exists which stores generated metadata and predefined domain ontologies. The users need to configure the initial URLs and the crawling-depth in the policy centre. Actual crawling process starts only after this. Metadata classifier will compute the similarities between the generated metadata and the ontology concepts. If the computed similarity is above a threshold value, the corresponding concept is regarded as being relevant to the metadata. Then, the metadata generator will associate the metadata with the concept. The semantic focused crawler prototype was run to download 2,000 business web pages under the category of transport in the Australian Yellowpages website. The results of evaluation are shown in Table 1. This crawler can deal with large-scale ontologies. But, since this crawler is built by combining the characteristics of ontology-based and metadata-based crawlers, it has all the disadvantages of both types of crawlers (Dong and Hussain, 2011).

2.6 Focused crawler based on link structure and content similarity

This crawler makes use of a combination of link-structure and similarity of contents between the web pages for performing the crawling. This crawler starts with a ‘seed’ which will be empty at the beginning. URLs from this page are fetched and stored in the database. Then in each step, one of the fetched pages will be added to the seed. The newly fetched pages are the ‘candidate crawled pages’. Attributes including similarity degree of the page to the domain, number of links from the page to seed pages and number of links from seed pages to the page are computed. These pages are ranked based on a metric which is a combination of these three attributes. In each stage, a page with highest rank will be added to the seed pages. Content similarity is a function which is based on the heuristic that usually similar words and phrases are used in pages belonging to a specific domain. For the purpose of evaluating this crawler, the developers ran it twice each time by giving an initial seed page. In the first run, the seed page was a search result page of Google for keyword sports and the second run was initiated with http://www.yahoo.com. The outputs of both the runs were compared with an ordinary BFS crawler. The major drawback of this crawler are if the initial Seed page does not relate to the domain, then the number of related pages will be very less in the beginning stages. This will affect the overall efficiency of the crawler (Jamali et al., 2006).

2.7 Board forum crawler

The main objective of a forum crawler is to retrieve as much user generated content and their associated information as possible. It also identifies and follows valuable links in the site. Most of the web forum sites are created by using forum software packages. So they will have an organised structure. Board forum crawler (BFC) exploits this characteristic of forum sites and simulates human behaviour of visiting forum sites. Crawling starts from the home page. It extracts board page seeds from homepage. After that for each board page seed, a link queue of all subsequent board pages in the same board is created. For each queue, each page in the queue is downloaded. Then the crawler checks whether it is exactly a board page and extracts links of post pages from the board page. A whole link index of all post pages in all board pages is created. Finally, post pages linked by the whole link index are downloaded. BFC was evaluated by using it to crawl a group of websites and the number of link-levels to be crawled was fixed as five. Precision and recall were calculated by using page downloaded (count of all pages downloaded), post downloaded (count of all post pages downloaded) and post all (count of all post pages). The drawbacks of this crawler are that the quality of the downloaded pages is not very high and also it will fail if the web pages have different structures (Guo et al., 2006).

Table 1 Assessment of existing web crawlers

Name	*Type*	*Features*	*Classification method*	*Algorithm*	*Evaluation results*
PyBOT	Generic	BFS, search tree and FIFO queue, shallow pages first, output in Excel CSV	NA	Breadth-first search	Crawling time (3 mins 38 secs)
Focused crawler	Focused	Focused on topic or domain, multithreaded	Keyword matching	NA	-
Semantic focused	Focused	Semantic web technology seed URL and topic description, priority assignment	Vectors	NA	Harvest rate (97%), precision (86.63%), recall (> 98%)
Ontology based	Focused	Ontology concepts, ontology base, multithreaded	Vectors	NA	-
Metadata abstraction	Focused	Annotation of web pages, metadata generator, mark-up languages	Naive Bayes, SVM, MSVM	NA	-
Metadata + Ontology	Focused	Ontology and metadata, annotation, classification	SVM or MSVM	NA	Crawling time-(216.92 s for 2,000 pages), harvest rate (97%), precision (86.63%), recall (> 98%)
Link structure+ Content similarity	Focused	Utilises link structure and content similarity seed pages, candidate crawled pages, ranking according to content similarity	NA	Clustering	Average harvest rate (> 60%)
BFC	Forum crawler	Developed for forums, three types of pages, input is homepage, uses a link queue	NA	BFC Algo.	Precision (avg 90%), recall (> 70%),
Link and text properties	Forum crawler	Text and link properties, URL clustering, traversal path (TP) generation, TP score computation	URL clustering	Traversal path score computation	Accuracy of cluster formation (100%), precision (up to 100%), recall (up to 100%), bandwidth saving (avg 62%)
iRobot	Forum crawler	Sitemap creation offline part for training, online part for crawling, algorithm is similar to prims	URL clustering	Minimum cost traversal path	Precision (70%), recall (> 85%),
FoCUS	Forum Crawler	Page types, URL types, ITF Regex, learning and crawling, URL detection	NA	URL type detection	Precision (avg 98%), recall (avg 95%)

2.8 Forum crawler based on links and text properties

The basic idea of this crawler is to use the outgoing links and text information in the forum pages. It consists of several modules. The URL sampling and clustering module categorises different types of pages into different clusters based on their URLs. A small number of pages from the targeted web forum are sampled randomly and all URLs in these pages are extracted. For each URL, a signature is generated by representing each character in the URL with a symbol and then combining consecutive similar symbols into a single symbol. URLs with the same signature are placed in the same cluster. A search for a set of keywords that can indicate different types of pages within each cluster is done. New clusters are formed which will better distinguish links based on the presence of the keywords within the links. The traversal strategy design module deals with the establishment of traversal relationship among the URL clusters and generation of candidate traversal paths. Traversal path scores are computed based on a traversal path score computation algorithm and the traversal path with the highest score is selected as the best traversal path. For evaluating this crawler, accuracy of cluster formation, precision and recall were used as parameters. To generate the traversal path for each forum site, 100 pages were randomly sampled from each site and subsequently all links in these sampled pages were clustered. Next, pages in each cluster were randomly sampled and several characteristics of these pages were computed. Finally, a traversal path was generated. A cluster was said to be correctly formed if the majority of the sampled targeted pages of the same type are grouped in the same cluster. A selected cluster in the traversal path was considered correct if it was in the correct order and correctly represented the corresponding targeted page. Bandwidth saving was calculated by calculating the percentage of redundant pages in the sampled pages. This crawler uses simple text and link properties in contrast to visual feature extraction which is costly. It is robust and is independent of the template of different pages in forums. But it has a drawback that best performance is obtained for forums with keyword-based URL structures only. The performance is negatively affected for forum sites with verbose-based URL structures (Sachan et al., 2012).

2.9 iRobot

The main idea of iRobot is to automatically rebuild the graphical architecture representation or the sitemap (Site maps, http://www.sitemaps.org/) of the target web forum site and then select an optimal traversal path which only traverses informative pages and skips invalid and duplicate ones. iRobot consists of two parts: an offline sitemap reconstructing and traversal path selection part and an online crawling part. The goal of the offline part is to mine useful knowledge from a few sampled pages. It is implemented using a double-ended queue and URLs are fetched randomly from the front or rear. Hence, the sampling process combines breadth first and depth-first strategies. Then, pages with similar layout are grouped into clusters according to the repetitive patterns they hold. This is carried out by the repetitive region-based clustering module. The outcome of this module is a list of repetitive patterns occurring in pages from the target forum. After that, according to their URL formats, pages in each cluster are further grouped into subsets by the URL-based sub-clustering module. Each subset contains pages with both uniform page layout and URL format and is taken as a vertex in the sitemap graph. The informativeness estimation module is responsible for selecting vertices with informative pages on the sitemap, and throwing away vertices with invalid or duplicate pages. The last module in the offline part is the traversal path selection module, whose function is to find out an optimal traversal path with minimum cost. After this, the online crawling part does the actual crawling by following the vertices in the constructed traversal path. For evaluating iRobot, seven different forums which were developed using different methods were selected. For each site, iRobot started from the homepage and followed the links belonging to that domain ignoring dulpicate URL addresses. These results were compared with a brute-force, a breadth first, and a depth-unlimited crawler. iRobot is more efficient when compared to other forum crawlers that were discussed above. But it cannot detect the entry URL on its own (Cai et al., 2008).

2.10 Forum crawler under supervision

Forum crawler under supervision (FoCUS) consists of two major parts: a learning part which learns ITF regexes (index-thread-page flipping regular expressions) of a given forum from URL training examples that are automatically constructed and an online crawling part which applies the learned ITF regexes for crawling. FoCUS finds the entry URL of a forum given any page of it using the entry URL discovery module. Entry URL discovery is based on several heuristics. Then the learning part of FoCUS starts. This part consists of two steps: constructing URL training sets which deals with the creation of highly precise index URLs, thread URLs and page-flipping URLs and learning ITF regexes from the training set. The index/thread URL detection module detects the index URLs and thread URLs on the entry page. SVM is used as the page-type classifier to identify which page a particular URL points to. The detected index URLs and thread URLs are stored in the URL training sets database. The destination pages of the detected index URLs are fed into this module again to detect more index and thread URLs until no more index URL is detected. After that, the page-flipping URL detection module tries to find page-flipping URLs from both index and thread pages and saves them also in the training sets. Page-flipping URLs are detected based on their layout characteristics. Finally, the ITF Regexes learning module learns a set of ITF regexes from the URL training sets. Four different regular expressions are generated for detecting the pages. After finishing the learning, FoCUS starts online crawling. It starts from the entry URL, follows all the URLs that are matching with the learned ITF regexes. Crawling continues until no page could be retrieved or a pre-defined condition is satisfied (Jiang et al., 2012). For the purpose of doing the evaluation, the developers selected 200 different forum software packages. For each software package, a forum developed by it was chosen. Among them, 40 forums were selected as training set and the remaining 160 were kept for testing. They have done a module-wise evaluation of FoCUS.

3 Evaluation criteria

Different web crawlers were evaluated using different parameters. The main four parameters were:

- *Precision*: Number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search or it is the fraction of retrieved documents that are relevant to the search.

  $$precision = (relevantpages \cap retrievedpages) / retrievedpages$$

- *Recall*: Number of relevant documents retrieved by a search divided by the total number of existing relevant documents or it is the fraction of the documents that are relevant to the query and that are successfully retrieved.

  $$recall = (relevantpages \cap retrievedpages) / relevantpages$$

- *Crawling time*: Time taken by the web crawler to crawl a pre-defined number of pages.
- *Harvest Rate (mainly used in focused crawling)*: Percentage of the web pages crawled which are related to the domain.

4 Assessment of web crawlers

After studying various web crawlers, it is observed that each one has its own features, advantages and limitations. Deep web crawlers retrieve many un-informative pages. Focused crawlers consume lesser amount of system resources compared to deep crawlers. None of them can be used for crawling web forums. Table 1 shows the summary of the findings that we obtained after reviewing almost all types of web crawlers that are existing today. The designers of each one evaluated them using different methods and based on different datasets. The results of their evaluations are also included in the table.

5 Research gaps

A web forum site or discussion board is an online site for sharing information and ideas through the internet. Most of the forum sites are developed using forum package softwares. So a web forum site has a specific structure when compared to normal websites. A web crawler designed for crawling a forum should follow this structure for effective crawling. A normal deep crawler cannot be used for this purpose since it is not capable of identifying the specific relation between the web pages in a forum site. Focused crawler can also be not used since it will retrieve pages based on a subject only. So for crawling web forums, special forum crawling softwares that are capable of following the forum structure have to be developed. One of the forum crawlers that was discussed above is board forum crawler or BFC. BFC works based on the idea that forum sites developed by the forum package softwares have similar structure. But this crawler yields less quality pages. BFC was developed exclusively for crawling Chinese web forum sites. Since these sites were developed using specific forum softwares, they had similar structures. BFC works well for such forum sites. But if the forum site contains pages which are not similar, BFC will fail to crawl. A method which will work irrespective of the specific structure of the forum sites should be developed.

Another forum crawler is the one which is based on the page text and link properties of web pages. This crawler considers the web forum site as having a hierarchical structure. This crawling strategy yields best performance only if the URLs in the forum site are keyword-based. If the URLs are verbose-based, then the performance will be reduced. A web forum crawler should work in the case of verbose-based and mixed type of URLs also. iRobot is a web crawler specially designed for forums. It is based on sitemap reconstruction and traversal path computation. But this crawler cannot detect the entry URL on its own. Crawling process starting from entry URL will give more coverage. FoCUS is a forum crawler which works based on URL classification and regular expression formation. This crawler gives better precision, accuracy and coverage as compared to other forum crawlers that were discussed above. Moreover, it has the capability to detect the entry URL of a forum site. But FoCUS itself has some drawbacks. First one is that it is using support vector machine (SVM) with a linear kernel setting for index/thread page classification. A better kernel setting has to be used for the page classifier which will give more accuracy and will take lesser amount of time to converge. FoCUS uses a weak page classifier along with majority voting method for index/thread URL detection. The issues related with this combined method are the following:

1 The outcome of a weak classifier may be erroneous.

2 If the URL group contains very less number of URLs (two or four) and if majority of them are misclassified, then that will affect the accuracy of the crawling process.

3 If the URL group contains a few URLs and if the number of URLs is even, then the majority voting method will fail if half of the URLs in the group are classified erroneously.

So the weak classifier-majority voting method combination has to be replaced. Similar to many other forum crawlers, FoCUS also is not capable of detecting JavaScript-based URLs. So if the page-flipping URL is JavaScript-generated, then only the first page will be retrieved and the remaining pages will not be crawled which negatively affects the precision and coverage. A new mechanism has to be used for detecting the JavaScript-based URLs. Many forum crawlers are using the page-features to classify the pages. Analysis of more and more forum pages will help to identify new features which will better distinguish the pages. Most of the forum crawlers including FoCUS use BFS for crawling. BFS has many disadvantages. So instead of using BFS, some other crawling strategy or combination of multiple strategies can be used.

6 Conclusions

In this paper, a review of different types of web crawlers is done. Different types of web crawlers are ordered as generic, focused and forum crawlers. One or more crawlers belonging to each type are reviewed. The advantages and disadvantages of each one are discussed. Finally, a table which contains an assessment of existing web crawlers is also given. After reviewing different crawlers, we could find that none of the deep web crawlers or focused crawlers can be used for crawling web forums since forum pages are different from normal web pages and among the four forum crawlers that were compared above, FoCUS yields very good precision and recall values. It also has an advantage that it is able to detect the entry URL of a web forum site on its own which is not done by others. This study will help researchers in understanding about different types of web crawlers. Also, it has explored the areas where more researches are to be done. A better web forum crawler can be developed which will overcome the drawbacks of existing crawlers and will perform the web forum crawling more efficiently.

References

Cai, R., Yang, J-M., Lai, W., Wang, Y. and Zhang, L. (2008) 'irobot: an intelligent crawler for web forums', in *Proceedings of the 17th International Conference on World Wide Web*, ACM, pp.447–456.

Dey, M.K., Chowdhury, H.M.S., Shamanta, D. and Ahmed, K.E.U. (2010) 'Focused web crawling: A framework for crawling of country based financial data', in *2010 2nd IEEE International Conference on Information and Financial Engineering (ICIFE)*, IEEE, pp.409–412.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C. and Sachs, J. (2004) 'Swoogle: a semantic web search and metadata engine', in *Proc. 13th ACM Conf. on Information and Knowledge Management*, pp.65–659.

Dong, H. and Hussain, F.K. (2011) 'Focused crawling for automatic service discovery, annotation, and classification in industrial digital ecosystems', *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 6, pp.2106–2116.

Guo, Y., Li, K., Zhang, K. and Zhang, G. (2006) 'Board forum crawling: a web crawling method for web forum', in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, pp.745–748.

Information retrieval parameters [online] http://http://en.wikipedia.org/wiki/Precision_and_recall (accessed 10 October 2013).

Internet forum [online] http://en.wikipedia.org/wiki/Internet_forums (accessed 15 September 2013).

Jamali, M., Sayyadi, H., Hariri, B.B. and Abolhassani, H. (2006) 'A method for focused crawling using combination of link structure and content similarity', in *IEEE/WIC/ACM International Conference on Web Intelligence, 2006, WI 2006*, IEEE, pp.753–756.

Jiang, J., Yu, N. and Lin, C-Y. (2012) 'Focus: learning to crawl web forums', in *Proceedings of the 21st International Conference Companion on World Wide Web*, ACM, pp.33–42.

Leng, A.G.K., Ravi, K.P., Singh, A.K. and Dash, R.K. (2011) 'Pybot: an algorithm for web crawling', in *2011 International Conference on Nanoscience, Technology and Societal Implications (NSTSI)*, IEEE, pp.1–6.

Maedche, A., Ehrig, M., Handschuh, S., Volz, R. and Stojanovic, L. (2002) 'Ontology-focused crawling of documents and relational metadata', in *Proceedings of the Eleventh International World Wide Web Conference WWW-2002*.

Najork, M. and Wiener, J.L. (2001) 'Breadth-first crawling yields high-quality pages', in *Proceedings of the 10th International Conference on World Wide Web*, ACM, pp.114–118.

Pinkerton, B. (1994) 'Finding what people want: Experiences with the web crawler', in *Proceedings of the Second International World Wide Web Conference*, Chicago, Vol. 94, pp.17–20.

Sachan, A., Lim, W-Y. and Thing, V.L.L. (2012) 'A generalized links and text properties based forum crawler', in *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, IEEE Computer Society, Vol. 1, pp.113–120.

Shah, U., Finin, T., Joshi, A., Cost, R.S. and Matfield, J. (2002) 'Information retrieval on the semantic web', in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ACM, pp.461–468.

Site maps [online] http://www.sitemaps.org/ (accessed 22 September 2013).

Yuvarani, M., Iyengar, N.Ch.S.N. and Kannan, A. (2006) 'Lscrawler: a framework for an enhanced focused web crawler based on link semantics', in *IEEE/WIC/ACM International Conference on Web Intelligence, 2006, WI 2006*, IEEE, pp.794–800.