A Survey of Focused Web Crawling Approaches

Sameendra Samarawickrama, Lakshman Jayaratne
University of Colombo School of Computing
35, Reid Avenue
Colombo 7, Sri Lanka
smsamrc@gmail.com, klj@ucsc.cmb.ac.lk

ABSTRACT: *In the last few years the volume and size of web has been growing exponentially making it difficult to search the related pieces of information. To ease such problem, we have various general as well as focused crawlers. These two types of crawlers have their own features and limitations. In this paper we provide an extensive survey of the available general and focused crawlers. Besides we support our work with experimental results. Additionally we provide classification of the crawlers based on the parameter, the text analysis. Based on this survey we also project the near future developments in crawler research.*

Keywords: Text Mining, Text categorization, Crawling, Web Crawler

Received: 11 October 2011, Revised 18 November 2011, Accepted 23 November 2011

1. Introduction

Once the web grows, it becomes important to introduce a search mechanism and the end users need to rely on such mechanism to enter into the world of information increase the popularity of web the among people. Web crawlers play a vital role inside a search engine such as finding new pages to be indexed, periodically recrawling and updating the index with fresh information etc. As of today the indexed web contains 17.5 billion web pages[1] and continues to grow at a rapid pace. With such a large web space, exhaustive searching becomes difficult with traditional general purpose web crawlers. It is also well known that general purpose search engines are not tailored at providing topic specific information. As a result vertical search engines (a.k.a. topical search engines) have gained its popularity in recent years, which are specifically designed to provide topic specific information. Focused crawler can be considered as the key component in a vertical search engine. It attempts to collect web pages relevant to a particular topic while filtering out the irrelevant. Thus focused crawling can be used to generate data for an individual user or community in their interested topic or automated building of web directories like Yahoo![2], Open Directory Project[3] which still uses human expertise for document categorization.

We have identified three major challenges a typical focused crawler meets: it needs to determine the relevance of a retrieved web page, predict and identify potential URLs that can lead to relevant pages, rank and order the relevant URLs so the crawler knows exactly what to follow next.

[1] http://www.worldwidewebsize.com/

[2] http://www.yahoo.com

[3] http://www.dmoz.org

The rest of this paper is organized as follows. Section II is about focused crawling terminology. A comprehensive survey of existing focused crawling approaches is carried out in Section III. Finally, Section IV discusses the important facts, future directions and make the conclusions.

2. Terminology

In the literature of focused crawling the term *harvest ratio* comes as the primary metric in evaluating the crawler's performance. It measures the rate at which relevant pages are fetched and how effectively irrelevant pages are kept out of the crawl. harvest ratio can be expressed as,

$$Harvest\ ratio = \frac{Relavent\ Pages}{Total\ pages\ downloaded}$$

It is always preferred high values for the harvest ratio.

Authorities are pages that are recognized as rich and relevant in content and *hubs* are pages with many links to other pages where those might contain relevant information. In-degree is one simple measure of authority. Jon Kleinberg - the introducer of these two terms - argued that hubs and authorities exhibit a mutually reinforcing relationship i.e., a good hub will point to many authorities and a good authority will be pointed at by many hubs. So a focused crawler should correctly exploit hubs to find authorities while sending relevant pages to be indexed.

crawler frontier contains the list of URLs to be crawled.

A page from which a link was extracted is called the *parent page* and the page pointed by the URL is called the *child page* or the *target page*. Some literature refers *target pages* to topic relevant pages.

Due to the complexity of web an irrelevant web page might refer to a highly relevant page. In this case the crawler has to traverse irrelevant page to get in relevant pages. This process is called the *tunneling*.

seed set is the set of URLs that are known to be highly relevant to the particular topic of interest. It is collected manually or by the crawler with the help of an existing search engine/portal.

Figure 1 shows how harvest ratios should normally vary for generic and focused crawlers. It is clear that focused crawlers always maintain a high harvest ratio than generic crawlers.

Figure 2 gives the high level overview of a simple focused crawler. URL downloader downloads web pages from WWW initiated with the seed URLs and sends them to the classifier. Classifier, which is trained with the help of seed set makes relevance judgments on pages it receives. URLs extracted from these relevant pages will be added to the crawler frontier to continue the

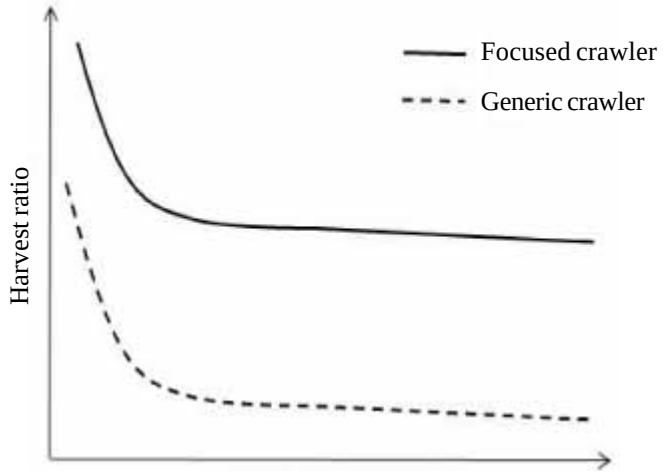

Figure 1. harvest ratio variation for generic and focused crawlers

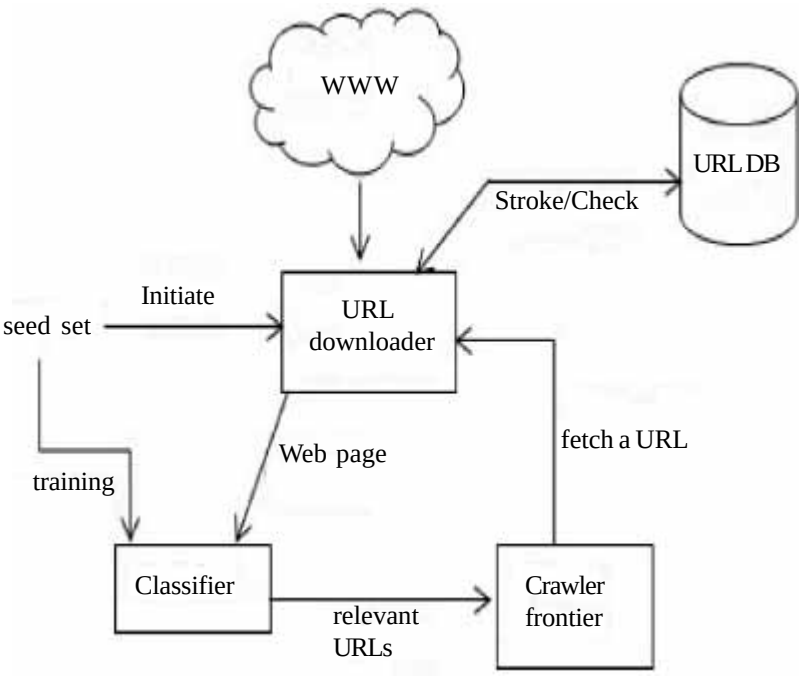

Figure 2. high level system overview

crawling process. URL downloader maintains a database (URL DB) of crawled pages. Upon retrieving a URL from the crawler frontier it first checks the URL DB to see that page has already been downloaded or not. URL DB may contain a local copy of the downloaded pages and will also serve the indexer as well.

3. Focused Crawling

Earliest work on focused crawling dealt with simple keyword matching, regular expression matching or using binary classifiers. A survey on different focused web crawling algorithms is presented in [17]. It categorizes algorithms as crawling without external help - like fish-search [4], Sharksearch [13] that make relevance judgments based on the text in the HTML anchor tag; crawling with the help of the background knowledge - that identifies relevant pages with the help of trained classifiers. This covers several algorithms up to 2003.

With the development of semantic web technologies like Web Ontology Language (OWL), query languages (e.g., SPARQL), semantic technologies inspired web has gained its popularity. These techniques have been adapted in to the focused crawling research as well. A survey on semantic technologies inspired focused crawlers can be found in [9].

In which they categorize them into three classes: ontologybased focused crawlers - that determine the relevance of web documents by analyzing the similarity between content and the ontology tree, metadata abstraction focused crawlers - that employ ontology mark-up languages to convert HTML documents into semantic web documents, other semantic focused crawlers - that have unique applications of semantic web technologies.

3.1 Classifier training in focused crawling

Classifier can be considered as the most important component of a focused crawler. Major task of it is to make relevance judgments on crawled web pages. It can be trained with supervised, unsupervised or any other learning paradigm.

Training data is obtained through a web portal, web search engine or a meta search engine. Usually this is a manual process. Term frequency (or its improved versions like TFIDF) of the document is used as feature (input) for the classifier. Optionally, techniques like latent semantic indexing (LSI) can be used to reduce the dimensionality of the feature vector before it is fed to the classifier.

Classifiers with supervised training require a set of labeled documents collected as described above, for its training. Naive Bayes, support vector machines, nearest neighbor and decision trees are to name some of the popular classifiers used in the literature. Based on experiments with four different classifiers - naive Bayes, nearest neighbor, decision trees, subspace method - Li and Jain [16] claim naive Bayes and subspace method outperform the other two classifiers on seven-class Yahoo news groups data sets. Also Gautam and Padmini [19] claim naive Bayes is a weak choice for a focused crawler when compared to support vector machines or neural networks. In our opinion we believe each classifier has its strengths and weaknesses and there is no hard and fast rule in deciding which classifier works best for what situation.

Classifiers with unsupervised learning mostly use similarity measures when making relevance judgments. Among others Cosine similarity measure can be considered as the most popular in the existing literature of focused crawling. For that, text is represented as a vector using some text representation model like vector space model. It is also required to create a vocabulary for the interested topic before using any similarity measures.

3.2 Focused crawling based on content analysis

Crawlers that are based on content analysis use document classification techniques to determine the relevance of a retrieved web page to a particular topic of interest, based on the page's content.

Assis et al. [8] describe an approach to focused crawling, which uses both the content-related information and genre information presented in web pages. Crawler analyzes a particular web page and identifies the terms that correspond to a genre and terms that are related to a specific topic, which a traditional crawler does not identify as separate. It also considers the page URL, which may contain terms related to both genre and the topic. Once the page is analyzed the Cosine similarity function is applied separately to each set of terms (the genre, topic (content) and the URL string) to generate a similarity score between the page and the specific topic of interest. If this score is greater than some threshold value that page is considered as relevant and hyperlinks will be added to the crawler frontier. Since this approach uses a simple set of heuristics to determine the relevance of a page as well as to guide the crawling, it does not require a training phase.

They conducted experiments in three subject areas - databases, data structures and information retrieval - with 15 terms defined as genre terms, 20 topic terms in each of the subject area and another 5 terms to represent the URL string showed that this crawler performs better than traditional classifier based crawlers. But this approach is not suited for domains, which can not be expressed by a set of genre terms.

Bazarganigilani et al. [3] propose a novel approach, which uses genetic programming (GP) to efficiently discover the best similarity function among Bag-of-words, Cosine and Okapi similarity measures. This is achieved with the fitness function used by the genetic algorithm.

Once a new page is fetched, the GP discovered best similarity measure will be used by the k-nearest neighbor classifier to determine the page relevance and will assign a similarity score for that page. This approach doesn't employ a method to predict the relevance of target URLs found in a page, rather it uses Breadth-First search to fetch all the outgoing links from a page and those URLs will be added to the crawler frontier. Although this can slow down the crawling process they claim crawling time is not a crucial factor in building vertical search engines since it is an offline process. The continuity of crawling process is determined by the decay value of that page. Decay values of child pages always decreases by half a factor of its parent's decay value (1, 1/2, 1/4, 1/8,...). This decay concept is applied when the crawler finds a page with decay value less than the decay threshold, then it checks to see if the similarity score of that page is greater than the reset threshold, if that is the case that page's decay value will be reset to one by the system.

Their experiments with seed URLs collected from a meta search engine showed GP based approach surpass both the content based support vector machine (SVM) and the combination based SVM approaches.

Das and Nandy [6] developed a hybrid rawler, which incorporates focused crawling principle as well as pipeline concept to add parallelism. Their approach towards the focused crawler is as follows. Crawler uses three key parameters: number of matches found in document section W_D, matches found in title tag W_T along with the number of words in the search query W_N, when calculating the page relevance. A page is classified into the document section and title section and weights are assigned according to the number of matches found in each section. Then the relevance of a page *D* to the topic *T* will be calculated using the below equation.

$$Relevance(D,T) = \frac{\sum W_N \theta}{\sum W_{D \in N} \phi + \sum W_{T \in N} \eta}$$

Here, θ, ϕ, η represents the weights assigned to W_N, W_D, W_T respectively. Since the topic is considered to provide the gist of the document title section is given a high weight and it is considered as $\eta = 2$.

Experiments were conducted in two areas - AIDS and Swine Flu - to show that this method is capable of fetching relevant pages until the crawling is stopped manually. However they did not compare this strategy against other focused crawling strategies in terms of the harvest rate.

A Simple Focused Crawler [21] is described by two parameters: degree of relatedness and depth. An exhaustive search is executed recursively (tunneling) up to a given degree of depth starting from the initial set of seeds in a chosen topic. From these set of pages, the page with the highest score, as assigned by the naive Bayes classifier will be selected with its "relatives" to proceed with the crawling process. Pages which are further and further away from this highest score page will be added to the seed set depending on the degree of relatedness, which is defined with respect to the highest score page.

Experiments for several topic areas tested showed that crawling up to a depth of two gives better harvest rates in the long run.

3.3 Focused crawling based on link analysis

In web crawling domain, link analysis is used to identify potential links that can lead to relevant pages. These focused crawlers utilize the concept of link's topical locality [7] in its crawling.

FOCUS [2] is based on the concept that the pages that are one link away are semantically more closer to the seed pages than the pages that are several links away. It uses this link distance to rank the pages directly rather than using the probability/relevance scores measured through a classifier.

They model the whole web as a layered graph, where the seed documents serves as the first layer and the pages that have links to first layer (i.e., to the seed set) as the second layer, the pages with links to the second layer as the third layer and so on. When traversing, layer two URLs will be given high priority over layer three URLs, which are closer to the chosen topic. Google SOAP Search API has been used for finding back-links for a page, creating the local neighborhood of the seed URLs.

They use an ordinal regressor to find the corresponding rank (layer) of a newly fetched document, which is initially trained with the layered graph created with the seed URLs. Whenever a newly fetched page is fed to the ordinal regressor, which will return a label indicating the predicted layer. Only the layers 1, 2, 3 are considered as relevant pages (layer 1 is equivalent the seed set) and others are simply discarded by the process.

Experimental results in topic areas - Mutual Funds, NASCAR, Soccer, and Cancer - which are considered as causing difficulties when crawling, showed that FOCUS out performs the baseline focus crawler (which decides page relevance by a simple binary keyword matching) giving better harvest rates.

Focused crawling using Navigational Rank (NR) [10] tries to address three disadvantages as they mention with regular focused crawlers: problems due to change of structure of different websites, not being able work well when seed pages are far away from topic relevant pages and ignorance of non-topic related pages, which can lead to related pages. They model all the pages of a website by a directed graph $G = (V,E)$ where node $v \in V$ represent a web page and and edge $e \in E$ represent a hyperlink. NR values are used to identify potential pages, which lead to topical pages and are calculated in two steps: calculation of first-step NR for a particular node *v, NR* (*v*) involves in-degree, relevance score of *v* as assigned by the naive Bayes classifier with other parameters; secondstep NR makes use of the already calculated first-step NR of that node.

Algorithm works as follows: first it downloads a set of web pages using Breadth-First search strategy and construct a web graph of these downloaded pages, then first-step NR is calculated for all the nodes in the graph to identify immediate important pages, with these, the calculated second-step NR scores are used to identify and download the possible topicrelevant pages.

NR tries to solve above mentioned three disadvantages by dynamically updating NR values for each node in the graph during the crawling process to adapt to different structures, considering the global connectivity of the graph rather than the local knowledge and by computing potential links, which lead to target pages by considering both the relevant and irrelevant pages (tunneling).

Their results showed NR outperforms two well known focused crawlers - reinforcement learning based focused crawler and context-graph based focused crawler - as well as Dynamic Personalized PageRank (DPPR) and Breadth-First Search.

3.4 Focused crawling based on content and link analysis

These crawlers identify relevant pages by the page content as well as try to analyze and predict the relevance of URLs found in identified relevant pages. So not all the URLs found in a page will be crawled by the crawler.

Focused crawler proposed by Anshika et al. [18] uses a general purpose search engine to build a topic specific weight table from the first few pages retrieved. Relevance of a page to the topic is computed using Cosine similarity measure with the help of a weight table constructed initially. Words inside HTML title tag will be given high priority over the other locations in the document. Unvisited URLs will be ranked using the information of pages that have been crawled and the metadata of the hyperlink (anchor text and href information).

It implements tunneling by keeping irrelevant pages in a separate data structure called "*irrelevant table*". Crawler simply follows these pages up to a certain degree as defined in the variable "*maxLevel*" in order to get relevant pages.

Experiments were carried in four domains - E-business, Nanotechnology, Politics, Sports - with 10 seed URLs for each, which were obtained using Google[4] and allowed the crawler to crawl up to 1000 web pages. Their results gave surprisingly higher harvest rates for this focused crawler as opposed to the Breadth-First search crawler and with the "*maxLevel*" set to 2. They haven't done any comparisons with focused crawlers.

HAWK [5] works as follows, fetch a page (initially a seed page or later a URL in the crawler frontier), calculate the relevance score of that page to the topic using HAWK's relevance score finding formula, if it is above some threshold value, the relevance scores for all the out-links will be predicted with Shark-search algorithm. For those URLs, whose predicted score is acceptable, will be prioritized using PageRank before adding to the crawler frontier.

Relevance score finding formula uses a topic vector, defined by experts, which contains (*keyword of topic, weight assigned for that keyword*) pairs, then it calculates the contribution of the document for each keyword of topic using the frequency each keyword appears in the document and the number of effective words in the document, which finally leads to the calculation of the relevance score.

Evaluation of HAWK against two other crawlers, Sharksearch and PageRank in various topics and upto 5000 crawls showed it outperforms the other two crawlers significantly with better harvest ratios.

Almpanidis et al. [1] presents a different approach, which minimizes the limitations imposed by the initial training data set. Used algorithm HCLA (Hypertext Content Latent Analysis) tries to combine content with the link analysis using the vector space model. Latent semantic classifier, which is trained with an unlabeled text corpus, analyzes both the link structure and the text content in deciding the page relevance with the help of HCLA. Use of an unlabeled text corpus makes the training process unsupervised and independent of historical information such as a previous crawl or an existing generalized search engine.

Experiments conducted comparing HCLA with five other techniques namely Breadth-First, BackLink count, PageRank, SS1 and SS2 which are two flavors of the Shark-search algorithm, showed that it yields better harvest ratios when the number of pages crawled is high. This might be due to lack of link information available during the initial stages of retrieval (since this is unsupervised training and thus no seed URLs to give high harvest ratios at the beginning). Both HCLA and PageRank requires high processing power and memory resources proved by the fact that HCLA was up to 100 times slower than the Breadth-First on some tests.

Qingyang et al. [23] proposed a focused crawler based on relational subgroup discovery technique and first-order logic[5]. The

[4] http://www.google.com

[5] first order logic is a more advanced, flexible version of prepositional logic

concept elements in a web page.

It uses predicates to model the relevance of unvisited URLs in the crawler frontier, which will be used by the relational learner to induce classification rules based on the first order logic. Six predicates are being used to decide the relevance in this way. For instance the predicate links_to (*Y, X, A*) denotes that the web page *Y* links to *X* through the anchor element *A*, the predicate *url_has*(*A*,*t*) means that anchor *A*'s *href* attribute contains the text token *t* and *text_has* (*E*,*t*) denotes that text token *t* occurs in one of the child text of element *E*. Those unvisited URLs that satisfy the rules will then be crawled based on the assigned priorities.

Their experiments with 10,000 seed pages and up to 10,000 crawls in four topic areas - Shakespeare, TeX, Python, Iraq War - showed better (surprisingly high for some topics) harvest ratios for first-order crawling over best-fit and accelerated crawlers.

BBF by C.Yin et al. [24] lets the user define the topic of their interest with informative terms provided by the system, which were extracted and ranked using x^2 statistics measure with the help of a general purpose search engine like Google. These informative terms discovered for each topic are considered as the topic keywords.

A crawled page is modeled as a vector (vector space model) before making any relevance judgments using the Cosine measure. This page relevance calculation takes in to consideration the number of topic keywords found in that page with other several parameters.

The relevance of target links will be predicted with the use of C4.5 decision tree and will be given a score - URLScore. C4.5 classifier uses words in the anchor text, words in the target URL, the link context (100 characters before and after the link) and the parent page's relevance to the topic. Those with good URLScore will be added to the crawler frontier ordered according to the URLScore.

They conducted experiments in four domains - sport, Chinese sport, music and American music - with the use of keywords obtained as described earlier. Starting from 10 seed URLs and crawling up to 1100 web pages, they showed that this approach BBF, outperformed both Best-First and Breadth- First in all four topics. The average harvest ratio for BBF in all 4 topics was 0.709 while it was 0.5 for Best-First and 0.1895 for the Breadth-First.

[11], [12] also use the same approach with different methods for calculating the URL Score and the weights of the topic keywords.

The proposed system of Wang et al. [22] contains three main parts: Classifier that estimates the relevance of a page, URL Queue, which contains the URLs to be crawled in an ordered way and the Page Downloader, which is an http client to download pages from WWW. Classifier is trained with an existing topic taxonomy using reinforcement learning paradigm. It involves preprocessing the web page i.e., checking the HTML content and deleting useless words, transforming URLs into canonical form; computing the weights of words found in link context (link context is the surrounding content pertaining to a particular URL) using an improved version of TF-IDF algorithm and carrying out relevance analysis of pages using the Bayes classifier. URLs extracted from topic relevant pages will be added to FIFO URL Queue according to their priority values estimated with the heuristic algorithm.

Their experimental results in various topics for up to 10000 page crawls showed this crawler performs better than both PageRank and Breadth-First search crawlers in terms of the harvest ratio.

User bookmarks can be a good and reliable source of getting information related to a particular topic of interest. Bingo! [20] utilizes these user bookmark files to build a hierarchical ontology, which will serve as seed URLs and training data for the classifier. Classifier is retrained periodically after a certain number of crawls to ensure a better classification. Bingo! uses HITS algorithm for link analysis and support vector machines for document classification.

4. Conclusions and Discussion

This paper surveyed several focused crawling approaches classifying them into three categories namely, crawlers based on

content analysis, crawlers based on link analysis and crawlers based on content and link analysis. These categories are not mutually exclusive and contains certain features common in all.

Crawlers based on content analysis make relevance judgments based on the content of retrieved web pages. Mostly they use unsupervised classifiers that use a similarity measure (cosine similarity is widely used) for determining the relevance of crawled pages. We observed they don't utilize methods to identify potential URLs and are greedy to crawl through all the URLs found in a relevant page thus implementing tunneling automatically. Most of them are designed to work for specific domains (health, IT). On the downside, these can waste an extra amount of storage and network bandwidth.

Crawlers based solely on link structure analysis ignore the page content when making its relevance judgments. Link structure analysis has its own problems due to the high dynamic nature of the web. Studies have shown that within a year, 80 percent of all links in the link structure will have changed or be new, 50 percent of all contents will be changed, 20 percent of web pages today will disappear [15].

Predicting link relevance is always difficult due to the small information pertaining to hyperlinks (link context). Out-link prediction is made using the keywords in the link context or an algorithm like Shark-search or by means of a classifier.

Crawlers that use supervised training paradigm should be very concern of their training data. Because the quality of training data for the classifier is crucial and can directly affect the effectiveness and the performance of the crawler. It can also be examined that these type of crawlers yield better harvest ratios initially than unsupervised crawlers due to the presence of seed URLs.

Apart from crawlers based on content analysis, other two are not domain specific, giving them high adaptability. Harvest ratio has been used as the primary evaluation metric across all types of crawlers. Most crawlers evaluate their performance with crawlers like Breadth-First, PageRank, BestFirst and show obvious performance boosts due to the fact that they are generic or novice focused crawlers. By comparison it is impossible as to say which crawling approach works best in what situation since they have been tested in varying test environments like topic domain, seed URLs, length of crawl etc.

Web pages are inherently noisy. People add artificial content into web pages in order to increase the rank in the search results. Also the use of advertisements and banners has dramatically increased within last few years. web site creators also use keyword spamming to improve the page's ranking.Focused crawler should be aware of these things and should adopt noise removal techniques, as these can be misleading the crawler.

Automatic text classification and focused crawling are highly inter-related. Most of the techniques used for text classification like probabilistic naive Bayes algorithm, reinforcement learning, neural networks, support vector machines have already been adapted in today's focused crawlers. We also believe that incorporating text classification with natural language processing techniques like Named Entity Recognition would boost the performance of the crawler. As it has already been proven to be effective in related research [14], but has not yet been tested with focused crawling.

References

[1] Almpanidis, G., Kotropoulos, C., Pitas, I. (2007). Combining text and link analysis for focused crawling - an application for vertical search engines. *Journal of Information Systems*, 32 (6) 886–908.

[2] Babaria, R., Nath, J. S., Krishnan, S., Sivaramakrishnan, K. R., Bhattacharyya, C., Murty, M. N. (2007). Focused crawling with scalable ordinal regression solvers. In *International Conference on Machine Learning*, p. 57–64.

[3] Bazarganigilani, M., Syed, A., Burkir, S. (2011). Focused web crawling using decay concept and genetic programming. *International Journal of Data Mining Knowledge Management Process (*IJDKP), 1.

[4] Bra, P. D., Jan Houben, H., Kornatzky, Y., Post, R. (1994). Information retrieval in distributed hypertexts. In *proceedings of the 4th RIAO Conference*, p. 481–491.

[5] Chen, X., Zhang, X., Hawk. (2008). A focused crawler with content and link analysis. *In*: *P*roceedings of the 2008 IEEE International Conference on e-Business Engineering, p. 677–680, Washington, DC, USA, IEEE Computer Society.

[6] Das, A., Nandy, S. (2010). Hybrid focused crawler - a fast retrieval of topic related web resource for domain specific searching. *International Journal of Information Technology and Knowledge Management*, 2, 355– 360.

[7] Davison, B. D. (2000). Topical locality in the web. *In*: Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000, p. 272–279. ACM Press.

[8] De Assis, G. T., Laender, A. H. F., Gonc alves, M. A., Da Silva, A. S. (2007). Exploiting genre in focused crawling. *In*: Proceedings of the 14th international conference on String processing and information retrieval, SPIRE'07, p. 62–73, Berlin, Heidelberg. Springer-Verlag.

[9] Dong, H., Hussain, F. H., Chang, E. (2009). State of the art in semantic focused crawlers. *In*: *Proceedings of the International Conference on Computational Science and Its Applications: Part II*, ICCSA '09, p. 910–924, Berlin, Heidelberg. Springer-Verlag.

[10] Feng, S., Zhang, L., Xiong, Y., Yao, C. (2010). Focused crawling using navigational rank. *In*: *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, p. 1513–1516, New York, NY, USA, ACM.

[11] Hati, D., Kumar, A. (2010). An approach for identifying URLs based on division score and link score in focused crawler. *International Journal of Computer Applications*, 2.

[12] Hati, D., Sahoo, B., Kumar, A. (2010). Adaptive focused crawling based on link analysis. *In*: *P*roceedings of 2nd International Conference on Education Technology and Computer (ICETC), p. 455–460.

[13] Hersovici, M., Jacovi, M., Maarek, Y., Pelleg, D., Shtalhaim, M., Ur, S. (1998). The shark-search algorithm - an application: tailored web site mapping. *In*: Proceedings of the 7th World Wide Web Conference.

[14] Hidalgo, J. M. G., Garc´1a, F. C., Sanz, E. P. (2005). Named entity recognition for web content filtering. *In*: Natural Language Processing and Information Systems, 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, Alicante, Spain, June 15-17, *In*: Proceedings, V. 3513 of Lecture Notes in Computer Scie*nce*, p, 286–297. Springer.

[15] Lew, D., Wahlig, H., Meyer-bautor, G. (2006). The freshness of web search engines databases.

[16] Li, Y. H., Jain, A. K. (1998). Classification of text documents. *The Computer Journal*, 41, 537–546.

[17] Novak, B. (2004). A survey of focused web crawling algorithms. *In*: Proceedings of SIKDD 2004 at multiconference IS.

[18] Pal, A., Tomar, D. S., Shrivastava, S. C. (2009). Effective focused crawling based on content and link structure analysis. *International Journal of Computer Science and Information Security (IJCSIS)*, 2, June.

[19] Pant, G., Srinivasan, P. (2005). Learning to crawl: Comparing classification schemes. *ACM Trans. Inf. Syst.*, 23, 430–462.

[20] Sizov, S., Siersdorfer, S., Theobald, Z., Weikum, G. (2002). The bingo! focused crawler: From bookmarks to archetypes. *In*: Proceedings of the 18th International Conference on Data Engineering, ICDE '02, Washington, DC, USA. IEEE Computer Society.

[21] Tsoi, A. C., Frosali, D., Gori, M., Hagenbuchner, M., Scarselli, F. (2003). A simple focused crawler. *In*: Proceedings of WWW 2003 (Posters).

[22] Wang, W., Chen, X., Zou, Y., Wang, H., Dai, Z. (2010). A focused crawler based on naive bayes classifier. *In*: *Third* International Symposium on Intelligent Information Technology and Security Informatics, p. 517– 521.

[23] Xu, Q., Zuo, W. (2007). First-order focused crawling. *In*: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, p. 1159–1160. ACM.

[24] Yin, C., Liu, J., Yang, C., Zhang, H. (2009). A novel method for crawler in domain-specific search, *Journal of Computational Information Systems*, p. 1749–1755.