State of the Art in Semantic Focused Crawlers

Hai Dong, Farookh Khadeer Hussain, Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology,
GPO Box U1987 Perth, Western Australia 6845, Australia
{hai.dong, farookh.hussain, elizabeth.chang}@cbs.curtin.edu.au

Abstract. Nowadays, the research of focused crawler approaches the field of semantic web, along with the appearance of increasing semantic web documents and the rapid development of ontology mark-up languages. Semantic focused crawlers are a series of focused crawlers enhanced by various semantic web technologies. In this paper, we make a survey in this research field. We discover eleven semantic focused crawlers from the existing literature, and classify them into three categories – ontology-based focused crawlers, metadata abstraction focused crawlers and other semantic focused crawlers. By means of a multi-dimensional comparison, we conclude the features of these crawlers and draw the overall state of the art of this field.

Keywords: focused crawlers, semantic web, semantic focused crawlers, ontology-based focused crawlers, metadata abstraction focused crawlers

1 Introduction

Semantic web is a vision for the future of the web in which information is categorized and made comprehensible by various automated tools [11].The major mission of the semantic web is to "*express meaning*" This demands that agents execute more intelligent operations on behalf of users [14]. A crawler is an agent which can automatically search and download webpages [4]. Focused (topical) crawlers are a group of distributed crawlers that specialize in certain specific topics [18]. Each crawler will analyze its topical boundary when fetching webpages.

Semantic web is an extension of World Wide Web with the purpose of expressing the meaning of the information [21] [22]. The technologies involved in the semantic web subsume (Extensible Markup Language) XML [24], XML schema [25], Resource Description Framework (RDF) [26], RDF schema [27], Web Ontology Language (OWL) [28] and SPARQL Protocol and RDF Query Language (SPARQL) [23] etc. Within them, the former four are employed to annotate web documents in order to convert them into semantic web documents; SPARQL is a RDF-based query language for querying the annotated documents [23].

Recently, focused crawler approaches the are increasingly being used in semantic web, along with the appearance of increasing semantic web documents and the rapid development of ontology mark-up languages [9] [10]. We define semantic focused crawlers *as a subset of focused crawlers enhanced by various semantic web* *technologies*. In this paper, we carry out a thorough survey from the literature on semantics-focused crawlers and provide directions for future and further research in this area. Eleven semantic focused crawlers are discovered and are classified into three groups. According to their features, we make a comprehensive evaluation of these crawlers along six dimensions, and thus draw conclusions in the final part.

2 Semantic Focused Crawlers

In accordance with the respective characteristics of the eleven semantic focused crawlers, we categorize them into three classes – ontology-based focused crawlers [19], metadata abstraction focused crawlers [20], and other semantic focused crawlers, which are defined in Table 1. From Section 2.1 to Section 2.3, we introduce the typical examples within these categories.

Table 1. Classification of semantic focused crawlers

Crawler category	Definition
Ontology-based focused crawlers	The focused crawlers that utilize ontologies to link a crawled web document with the ontological concepts (topics), with the purpose of organizing and categorizing web documents, or filtering irrelevant webpages with regards to the topics [19].
Metadata abstraction focused crawlers	The focused crawlers that can abstract and annotate metadata from the fetched web documents, in addition to fetching relevant documents [20].
Other semantic focused crawlers	The focused crawlers that employ other semantic web technologies than ontology-based filtering and metadata abstraction.

2.1 Ontology-based Focused Crawlers

In the existing literature there are four ontology-based focused crawlers, namely (a) LSCrawler (b) Courseware Watchdog crawler (c) crawler proposed by Ganesh et al. (d) THESUS crawler. In this section, we present an overview of each of these four crawlers.

Yuvarani et al. [16] propose a new generation of focused crawler – LSCrawler – which makes use of ontologies to analyze the semantic similarity between URLs and topics. In the LSCrawler, an ontology base is built to store ontologies. For each query keyword, a Relevant Ontology Extractor retrieves the ontology base to find the compatible ontology. Then the matched ontology is passed to a Crawler Manager. Meanwhile, a Seed Detector sends the keyword to the three most popular search engines, and returns the retrieved seed Uniform Resource Locator (URLs) to the URL Buffer of the Crawler Manger. Based on the matched ontology and the retrieved URLs, the Crawl Manager then generates a multi-threaded crawler to fetch webpages by these URLs. Meanwhile, a Busy Server is configured to prevent repeatedly visiting URLs, which have already been visited. The fetched webpages are then stored into a Document Repository, and the fetched URL database is updated. Subsequently, a

Link Extractor extracts all URLs and their surrounding texts from the fetched webpages, and sends them to a Hypertext Analyzer. Meanwhile the Porter Stemmer algorithm is used to remove stop keywords and extract terms from the texts. The Hypertext Analyzer then removes the URLs found in the fetched URL Database, and the extracted terms are matched with the concepts in the ontology, to determine the relevance of webpages to the keyword. Based on the relevance values, the URLs are ranked and then stored in the URL repository for further visit. In order to evaluate the framework of the proposed LSCrawler, the authors compare the performance of LSCrawler-based search engine with a full text-indexed search engine on the benchmark of recall. The results show that the LSCrawler has a 10% advantage on average.

Tane et al. [15] propose a new ontology management system – Courseware Watchdog. One important component of the system is an ontology-based focused crawler. By means of the crawler, a user can specify his/her preference, by assigning weights to the concepts of an ontology. By means of the interrelations between concepts within the ontology, the weights of other concepts can be calculated. Once a webpage is fetched, its text and URL descriptions are matched with the weighted ontological concepts. Thus, the weights of the webpage and its URLs are measured, ranked and clustered according to the concepts. In addition, the webpage relations can be viewed by linking the webpages to the ontology concepts that appear in the webpages.

Ganesh et al. [6] propose a group of metrics, with the purpose of optimizing the order of visited URLs for web crawlers. Three metrics are involved, which are combination importance metric, association metric and ordering metric. First of all, given a webpage *p*, the combination importance metric *CI(p)* can be computed as shown below:

$$CI(p) = a1 \cdot IB(p) + a2 \cdot IL(p) + a3 \cdot IF(p) + a4 \cdot IR(p) \tag{1}$$

where *a1*, *a2*, *a3* and *a4* are constants, *IB(p)* is the number of inbound links to webpage *p*, *IL(p)* is the location weight of webpage *p*, *IF(p)* is the number of outbound links from webpage *p*, and *IR(P)* is the PageRank [30, 31] weight of webpage *p*, which can be computed by (2) shown below:

$$IR(p) = (1-d) + d \sum_{p_i \in IB(p)} IR(p_i) / IF(p_i) \tag{2}$$

where *d* is a damping factor.

For each URL u_i in the webpage *p*, two association metrics $AS(u_i)$ and *AS(P)* respectively evaluate the semantic relevance of the URL u_i and the semantic relevance of webpage *p* based on a reference domain-specific ontology. These association metrics can analyze the link strength between parent and children webpages after the latter is downloaded, in order to refine itself. Finally, the URLs would be ranked according to an ordering metric *O(u)* that can be mathematically expressed below:

$$O(u) = b1 \cdot CI'(p) + b2 \cdot AS(u) + b3 \cdot \sum_{p_i \in P(u)} AS(p_i) \tag{3}$$

where b1, b2, b3 are constants, CI'(p) is the combination importance metric that evaluates the downloaded webpage *p*, *P(u)* is a function that returns all the parent pages of a URL *u*.

THESUS aims to organize online documents by linking their URLs to hierarchical ontology concepts, which are seen as thematic subsets [8]. A web crawler is used in the document acquisition component of the system. The working mechanism of this crawler is as follows: first, the crawler extracts the URLs and their descriptive texts from the initial set of documents; then the descriptive text of a URL are matched with one of the ontological concepts, and the URL is linked to concept. A threshold of maximum number of recursions or maximum number of documents is set in order to ensure that the process is not carried out endlessly. For a web document d_j, the crawler extracts a set of terms k_i (i = 1...n) with the respective weights n_i extracted from the descriptive texts of the URLs pointing to the document. Then the document d_j can be seen as the following set as: {URL, k_i, n_i} (i = 1...n). The similarity $sim_{ct,ki}$ between a concept c_t and terms k_i (i = 1...n) is computed as shown below:

$$sim_{c_t,k_i} = \frac{\sum_{L(k_i,c_t)} n_i \cdot s_t}{\sum_{L(k_i,c_t)} n_i} \tag{4}$$

where $L(k_i, c_t)$ is a function that returns all available couplings between terms k_i (i = 1...n) and concept c_t, s_t is the indication-based weight of each concept c_t.

Therefore, the web document d_j then can be seen as a following set as: {URL, c_t, $sim_{ct,ki}$} (t = 1...m), which can be utilized for the following ontology-based document clustering.

In order to evaluate the crawler's framework, the authors compare its clustering efficiency with a keyword-based clustering approach based on the benchmarks of F-measure, rand statistics, preprocessing time and average clustering time. The result indicates that the THESUS crawler has a 0.12 advantage on F-measure, a 0.05 advantage on rand statistics, over 40 times faster on preprocessing time and 0.4s faster on average clustering time than the latter.

2.2 Metadata Abstraction Focused Crawlers

In the existing literature there are two metadata abstraction focused crawlers namely Vertical Portal crawler and CiteSeer crawler. In this section, we present an overview of these crawlers.

Francesconi and Peruginelli [5] propose Vertical Portal, with the purpose of providing both resources and available solutions and services to satisfy users' requirements, within the legal domain. In the system a metadata abstraction focused crawler is designed by the authors, to fetch the domain-specific web documents. Then a metadata generator automatically transforms the web documents into metadata, by means of extraction. The focused crawler is implemented by computing the probabilities that URLs are similar with predefined topics. The metadata format is in accordance with the Dublin Core (DC) scheme in its XML version. Then, with the purpose of document clustering, each document *d* can be represented as a vector of weights ($w_1, \ldots, w_n$), in which each weight could be one of the following three types:

- Binary weight $\delta(w, d)$ that indicates the presence/absence of a term *w* in a document.
- Term frequency (tf) weight $tf(w, d)$ that indicates the frequency of a term *w* appearing in a document.
- Term frequency-inverse documents frequency (tfidf) weight $tfidf(w, d)$ which can be mathematically computed, as shown below:

$$a_{ij} = \frac{freq}{\max freq} \bullet \log \frac{N}{n} \tag{5}$$

where *freq* is frequency of term *w* appearing in a document, *maxfreq* is the total number of terms appearing in the document, *N* is total number of documents and *n* is the number of documents where term *w* appears.

During the next stage, two document classification algorithms – Naive Bayes (NB) and Multiclass Support Vector Machines (MSVM) are adopted respectively for evaluation purposes. The results show that the latter has a 2.6% advantage than the former on accuracy.

Giles et al. [7] propose a niche search engine for retrieving e-business information, with the integration of CiteSeer technique. A set of crawling strategies, including Brute Force, Inquirus-based and focused crawlers are used to fetch web documents. The CiteSeer technique is used to parse citations from the downloaded documents, and then create metadata based on the documents. To enhance the quality of metadata, the Support Vector Machine (SVM) algorithm is chosen to extract metadata, in comparison with the Hidden Markov Model (HMM) algorithm. Based on a small training set of words, the SVM model shows better performance than the HMM on accuracy.

2.3 Other Semantic Focused Crawlers

The following focused crawlers all have their own unique features that differentiate them from the previously discussed ontology-based crawlers and metadata abstraction crawlers, and therefore cannot be grouped them. From the literature, there are five such crawlers, namely Lokman crawler, the crawler proposed by Liu et al., Web Spider, Digital Library crawler and BioCrawler,

Can and Baykal [2] propose a medical search engine – MedicoPort – which employs a topical web crawler – Lokman. Lokman is responsible for collecting the medical information while limiting the scope of linking URLs. By means of the concepts from a Unified Medical Language System (UMLS) [29], Lokman can identify the links relevant to the medical domain. For each fetched document, a Document Parser extracts the links from it. For each fetched webpage, its relevance value to the UMLS concepts are estimated based on the concept frequencies, concept weights, and the relevance value of the contained URLs. The URL relevance values are evaluated by a Link Estimator, based on the relevance between the texts within the URLs and the UMLS concepts. Then a URL Frontier determines the order of URL queue, based on their relevance values. Lokman then fetches the URLs within the URL queue. The performance of Lokman is tested by the comparison between two re-evaluation algorithms – IncrementValues which regards the sum of link relevance values for a link as its link value, and GetGreater which regards the maximum value as the link value. Two situations, which are direct links included and excluded out of the seed URLs, are tested by means of the two re-evaluation algorithms. In comparison with a simple best search crawler, Lokman shows significant improvement in both the situations.

Liu et al. [12] propose a learned user model-based approach to assist focused crawlers to predict relevant links based on users' preference. Three components are involved in their architecture, which are User Modeling, Pattern Learning, and Focused Crawling. In the User Modeling, the system observes the sequence of user-visited pages with regards to a specific topic. A web graph is drawn, which consists of nodes that represent the webpages user-visited, and edges that represent the links among the webpages, in order to analyze user browsing pattern. In addition, the nodes are highlighted when users regard them as relevant. In the pattern leaning, the Latent Semantic Indexing (LSI) model is adopted to cluster the documents to several groups, and to reveal the topic for each cluster and the relationship between the identified topics. Meanwhile an Expectation-Maximization (EM) algorithm is used to optimize the clusters. Then a Hidden Markov Model (HMM) algorithm is used to estimate the likelihood of the topics directly or indirectly leading to a target topic. The mathematical representation of the HMM [32] is described as follows:

Let $S = (T_{n-1}...T_0)$ be n hidden states of reaching an object, O = $(o_1...o_m)$ are m visible states associated with two conditional probability distributions $P(s_j|s_i)$ and $P(o|s_j)$, then the Initial Probability Distribution Matrix $P = \{P(T_0)... P(T_{n-1})\}$, the Transition Probabilities Matrix $A = [a_{ij}]_{n\times n}$, where a_{ij} = probability of being in state T_j at time $t+1$, given that the observer is in state T_i at time t, and Emission Probabilities Matrix B = $[b_{ij}]_{n\times m}$, where b_{ij} = probability of seeing cluster j if the observer is in state T_i. The probabilities are estimated by maximum likelihood with ratio of counts, which can be shown mathematically as follows:

$$a_{ij} = \frac{|L_{ij}|}{\sum_{k=0}^{n-1} |L_{kj}|} \tag{6}$$

$$b_{ij} = \frac{|N_{ij}|}{\sum_{k=1}^{m} |N_{kj}|} \tag{7}$$

where $L_{ij} = \{v \in T_i, w \in T_j : (v,w) \in E\}$ and $N_{ij} = \{C_i : C_i \in T_j\}$.

In the focused crawler, the HMM is used to find the most likely state sequence in state s at time $t+1$ given the observed webpage sequence which can be mathematically shown below:

$$\delta(s,t+1) = \max_{s'} \delta(s,t) P(s \mid s') P(o_{t+1} \mid s) \tag{8}$$

where $max\delta(s, t)$ is the maximum probability of all sequences ending at state s at time t, $P(s|s')$ and $P(o_{t+1}|s)$ are transition probabilities and emission probabilities.

In the Focused Crawling, a focused crawler downloads the page linked to the first URL in its URL queue, and computes the page's reduced LSI representation. It then downloads all the children pages and clusters them by means of the K-Nearest Neighborhood algorithm to obtain the corresponding Visit Priority Value based on the learned HMM. In comparison with a Best-First search crawler, the crawler shows significant advantage on precision [13].

Cesarano et al. [3] propose an agent-based semantic search engine. In their proposed methodology, the query keywords are sent to a traditional search engine and the retrieved URLs are returned. One of the components of the search engine – Web Spider – can download all pages by URLs and then visit all children pages pointed by the URL, which traditional search engines can not reach. The web spider uses a Web Catcher which follows links to visit web pages. Then the web pages are stored in a Web Repository, and the unvisited links parsed from the web pages are visited next time. The whole crawling procedure stops when a predefined depth parameter is reached. Then a Document Postprocessor extracts the useful information for each downloaded page, including the title, content and description; a Miner Agent ranks these pages according to the similarities between the pages' information and a user-predefined search context. The tool used for computing similarity values is the group of ontologies stored in a Semantic Knowledge Base, which has weighted relations between concepts. The similarity between concepts c_i and c_j are obtained by (9) shown below:

$$d(c_i, c_j) = \max_{i \in (1..n)} \left(\prod_{j=1}^{m_i} P_{ij} \right) \tag{9}$$

where n is the number of existing paths between c_i and c_j, m_i is the number of edges in these paths and P_{ij} is the weight on each edge. The semantic relevance of a webpage is considered to be a function of single-word concepts, which consists of the following processes:

- The title, content and description of a webpage are extracted as a sequence of concepts.
- The Normalized Probabilistic Distance between each pair of concepts can be computed by (10) shown below:

  $$NPD(c_i, c_j) = \frac{d(c_i, c_j)}{DIST(c_i, c_j)} \tag{10}$$

  where $DIST(c_i, c_j)$ is the distance between the words representing concept c_i and c_j.
- The Semantic Grade of a webpage can be computed by (11) shown below:

  $$SeG = \sum_{h=1}^{NC} \sum_{k=h+1}^{NC} NPD(C_{ik}, C_{ih}) \tag{11}$$

  where NC is the number of concepts appearing in the webpage.

- Then the Semantic Grade is normalized by (12) shown below:

$$NSeG_i = \sum_{h=1}^{NC} \sum_{k=h+1}^{NC} \frac{2 \cdot NPD(C_{ik}, C_{ih})}{NC^2 + NC} \qquad \textbf{(12)}$$

- Finally, the Normalized Semantic Grade for a webpage can be mathematically shown as follows:

$$NSeG = \sum_{i \in (t,c,k,d)} \rho_i \cdot NSeG_i \qquad \textbf{(13)}$$

  where t is the title of the webpage, c is the content of the webpage, k is the keywords of the webpage, d is the description of the webpage and $\rho_t + \rho_c + \rho_k + \rho_d = 1$.

Zhuang et al. [17] propose to use publication metadata to guide focused crawlers to collect the missing information in digital libraries. The whole procedure is as follows: when a request for retrieving the publications in a specific venue is sent by a user; a Homepage Aggregator queries a public metadata repository, and returns some Metadata Heuristics for a focused crawler to locate the authors' homepages, it also returns a list of URLs to a Homepage URL Database; then the focused crawler fetches the publications by means of the seed URLs and stores them in a Document Database.

Batzios et al. [1] propose a vision of crawler – BioCrawler – working in the environment of semantic web. BioCrawler extends from the focused crawler, which is a group of distributed crawlers over the web, which is seen as an entity of "*vision, moving, communication abilities*", and an up-to-date knowledge model when browsing web content. Vision is the scope of domains which one BioCrawler can visit, in the form of web page link vectors. Thus, BioCrawler's movement is controlled by their visions. A Rule Manager agent is configured to determine the best rule (route) upon a crawler's request, based on the strength parameter of each available route plan. The knowledge model mechanism in BioCrawler is composed of a classifier that stores the information regarding rules, and a classifier evaluator which calculates the amount of semantic content grabbed by following the rules, also called the rules' strength. In order to evaluate its framework, the BioCralwer is compared with a dump crawler on the benchmark of crawler energy that is defined as the amount of webpages crawled per unit of bandwidth. Two experiments are implemented, which respectively compare the crawler energy of the two crawlers during 30,000 website visited and their average crawler energy during 100 random re-starts. Both of the experiment results show that the BioCrawler outperforms the dump crawler.

3 Comparison of the Semantic focused Crawlers

In the following sections, we make a comprehensive comparison to the introduced semantic focused crawlers by their categories. Based on their typical features, we choose the seven dimensions below for comparison: domain of application (e.g., business, medicine etc.), working environment (e.g., Google crawler for Google, Yahoo crawler for Yahoo etc.), special functions, technologies utilized, evaluation methods, evaluation results, and finally our comments or suggestions to the crawlers.

3.1 Comparison of the Ontology-based Focused Crawlers

We would like to observe that most of the ontology-based focused crawlers are designed for general domains. Some of them are encapsulated in larger systems, and others are designed as separate tools. Ontology is mostly used to match the fetched URLs or webpages with the predefined topics (ontological concepts), by means of computing the similarity value between the ontological concepts and the fetched URLs or webpages. In addition, the weight of some ontological concepts can be defined by users to highlight users' preference. While most of the crawlers do not provide evaluation methods and results, those that provide make use of precision and recall as primary metrics to measure the performance of such crawlers. As a whole, the ontology-based focused crawlers show obvious progress, compared with some full-text crawlers. We suggest that the crawlers' designers should provide more technical details regarding their evaluation process and results, in order to consolidate the crawlers. The detailed comparison results of the ontology-based crawlers are shown in Table 2 (see appendix).

3.2 Comparison of the Metadata Abstraction Focused Crawlers

From this comparison it is found that the metadata abstraction crawlers mostly work in the specific domains and capsulated in more comprehensive systems. Due to the specialty of documents fetched, they need to convert the domain-specific document into more meaningful metadata. Various technologies are utilized for document classification and metadata abstraction. Whilst some crawlers do not provide their evaluation details and results, from the existing evidence based on the preliminary survey, we still can observe its prime performance. It is suggested that the authors should disclose their evaluation details and results. The detailed comparison results can be found in Table 3 (see appendix).

3.3 Comparison of the Other Semantic Focused Crawlers

The ungrouped crawlers display the flexibility of the application of semantic web technologies in focused crawlers. Most of the crawlers are applied as part of a larger system in a specific domain, such as search engine, knowledge portal and so forth. Differing from the traditional focused crawlers, they have some special functions, such as estimating the similarity values between documents/URLs and ontology concepts/user-predefined context, indexing the unvisited URLs based on ontology concepts and users' preference, seeking the missing documents in a metadata base. Multiple semantic web technologies are used, including ontologies, similarity and clustering algorithms, as well as metadata heuristics. The primarily used evaluation method is to compare the harvest rate or precision with a Best-First Crawler or

Breadth-First Crawler and nearly all of them show significant advantages. However, the disadvantages are also obvious: some crawlers contain many complex algorithms or operations which may affect their efficiency; some need more testing, considering the obvious variation in different environments; some should provide metadata abstraction function for the interdisciplinary knowledge sharing. The detailed comparison results can be found in Table 4 and 5 (see appendix).

3.4 Conclusions and Recommendations Toward the Comparison of Semantic Focused Crawlers

From the respective comparison of three clusters of crawlers, the conclusions with regard to the features and situation of the semantic focused crawlers are made as follows:

First of all, let us emphasize on the features of each category of crawlers with semantic web technologies. For the ontology-based focused crawlers, the utilization of semantic web technologies mainly focuses on the use of ontology for linking webpages/URLs with topics (ontological concepts), indexing webpages based on estimating the similarity values between webpages and ontological concepts, or analyzing users' preference, in order to provide personalized crawling services; for the metadata abstraction focused crawlers, the utilization focuses on annotating the parsed and extracted web information with the ontology mark-up languages; for the category of other semantic focused crawlers, ontology can be used to calculate the similarity values between webpages or between webpages and queries etc..

Next we conclude the comparison results of the semantic focused crawlers from the dimension of domain, working environment, evaluation method and result. The domains where the semantic focused crawlers work can be divided into two categories – general and specific domain. They are designed either as a part of a complex system, such as search engine, or only as a tool that can be used independently or as a plug-in for any systems. The evaluation methods focus on the traditional methods in information retrieval – precision and recall. By means of the comparison with the traditional full-text, Best-First or Breadth-First crawlers, the significant advantages of the series of semantic crawlers are indicated directly. However, apart from the advantages, the disadvantages are obvious – many proposed models are not tested, which reveals that this field is not mature yet.

In the backdrop of these semantic focused crawler shortcomings, we recommend that these researchers should disclose their evaluation details and compare their crawlers with other crawlers without semantic technological supports, in order to validate the feasibility and applicability of their research.

4 Conclusion

In this paper, we carried out a detailed survey in the field of semantic focused crawlers. According to the literature, we classify the existing semantic focused crawlers into three primary categories – ontology-based focused crawlers that determine the relevance of web documents by analyzing their relevance to ontology concepts, metadata abstraction focused crawlers that employ ontology mark-up languages to convert HTML documents into semantic web documents, and other semantic focused crawlers that have unique applications of semantic web technologies. Based on a thorough literature analysis, we found eleven crawlers in this domain. The working mechanism of each of these research crawlers is explained in detail.

In order to perform a comparative analysis of these crawlers, for each category of crawlers, identified six key attributes for comparison and evaluation purposes. These are the *domain, working environment, special functions, technologies utilized, evaluation methods* and *evaluation results*. We observed that the ontology-based focused crawlers focus on using ontology for linking webpages/URLs with topics (ontological concepts), indexing webpages based on estimating the similarity values between webpages and ontological concepts, or analyzing users' preference, in order to provide personalized crawling services. Additionally, we observed that the metadata abstraction focused crawlers focuses on annotating the parsed and extracted web information with the ontology mark-up languages; other semantic focused crawlers employ ontology to calculate the similarity values between webpages or between webpages and queries etc. By means of this comparison, we came to the conclusion that these semantic focused crawlers have significant advantages in contrast to the traditional crawlers. However, some researchers do not disclose their evaluation details and results, which indicate the "*blueprint*" stage of the semantic focused crawler research.

In conclusion, on one hand, the application of semantic web technologies achieves undebatable progress in the field of focused crawler research; on the other hand, it is still far away to claim success to researchers, which reveals the state of the art in this field.

References

1. Batzios, A., Dimou, C., Symeonidis, A.L., Mitkas, P.A.: BioCrawler: An intelligent crawler for the semantic web. Expert Systems with Applications **35** (2008) 524-530
2. Can, A.B., Baykal, N.: MedicoPort: A medical search engine for all. Computer Methods and Programs in Biomedicine **86** (2007) 73-86
3. Cesarano, C., d'Acierno, A., Picariello, A.: An intelligent search agent system for semantic information retrieval on the internet WIDM '03. ACM Press, New Orleans (2003) 111-117
4. Cho, J., Garcia-Molina, H.: Parallel Crawlers. WWW2002. ACM Press, Honolulu (2002) 124-135
5. Francesconi, E., Peruginelli, G.: Searching and retrieving legal literature through automated semantic indexing. ICAIL '07. ACM Press, Standford (2007) 131-138
6. Ganesh, S., Jayaraj, M., Kalyan, V., Aghila, G.: Ontology-based web crawler. The International Conference on Information Technology: Coding and Computing (ITCC'04). IEEE CS, Las Vegas (2004)
7. Giles, C.L., Petinot, Y., Teregowda, P.B., Han, H., Lawrence, S., Rangaswamy, A., Pal, N.: eBizSearch: A niche search engine for e-business. SIGIR'03. ACM Press, Toronto (2003) 213-214
8. Halkidi, M., Nguyen, B., Varlamis, I., Vazirgiannis, M.: THESUS: organizing web document collections based on link semantics. The VLDB Journal **12** (2003) 320–332

9. Hendler, J.: Agents and the semantic web. IEEE Intelligent System **16** (2001) 30-37
10. Jansen, B.J., Mullen, T., Spink, A., Pedersen, J.: Automated fathering of web information: an in-depth examination of agents interacting with search engines. ACM Transactions on Internet Technology **6** (2006) 442-464
11. Konopnicki, D., Shmueli, O.: Database-inspired search. The 31st VLDB Conference, Trondheim (2005) 2-12
12. Liu, H., Milios, E., Janssen, J.: Focused crawling by learning HMM from user's topic-specific browsing. The IEEE/WIC/ACM International Conference on Web Intelligence (WI'04). IEEE CS (2004)
13. Liu, H., Milios, E., Janssen, J.: Probabilistic models for focused web crawling. WIDM'04. ACM Press, Washington, DC (2004) 16-22
14. Stojanovic, L., Stojanovic, N., Volz, R.: Migrating data-intensive web sites into the semantic web. SAC 2002. ACM Press, Madrid (2002) 1100-1107
15. Tane, J., Schmitz, C., Stumme, G.: Semantic resource management for the web: an e-learning application. WWW2004. ACM, New York (2004)
16. Yuvarani, M., Iyengar, N.C.S.N., Kannan, A.: LSCrawler: a framework for an enhanced focused web crawler based on link semantics. The 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06). IEEE (2006)
17. Zhuang, Z., Wagle, R., Giles, C.L.: What's there and what's not? : Focused crawling for missing documents in digital libraries. JCDL'05. ACM Press, Denver (2005) 301-310
18. Barfourosh, A.A., Anderson, M.L., Nezhad, H.R.M., Perlis, D.: Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition. Department of Computer Science, University of Maryland, Maryland (2002) 1-45
19. Dong, H., Hussain, F.K., Chang, E.: A survey in semantic web technologies-inspired focused crawlers. The Third International Conference on Digital Information Management 2008 (ICDIM 2008). IEEE, East London (2008)
20. Dong, H., Hussain, F.K., Chang, E.: State of the art in metadata abstraction crawlers. 2008 IEEE International Conference on Industrial Technology (IEEE ICIT 2008). IEEE, Chengdu (2008)
21. W3C Semantic Web Frequently Asked Questions. W3C (2008)
22. Berners-Lee, T.: The semantic web. Scientific American Magazine, Vol. May 17 (2001)
23. Rapoza, J.: SPARQL will make the web Shine. eWeek, Vol. May 2 (2007)
24. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F.: Extensible Markup Language (XML) 1.0 (Fourth Edition)-origin and goals. W3C (2006)
25. Sperberg-McQueen, C.M., Thompson, H.: XML Schema 1.1. W3C (2000)
26. Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): concepts and abstract syntax. In: McBride, B. (ed.): W3C Recommendation W3C (2004)
27. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. In: McBride, B. (ed.): W3C Recommendation. W3C (2004)
28. Herman, I.: Web Ontology Language (OWL). W3C (2007)
29. Unified medical language system. National Library of Medicine (2008)
30. Cho, J., Garcia-Molina, H., Page, L.: Efficient crawling through URL ordering. the Seventh International World Wide Web Conference (WWW' 98). ACM (1998) 161-172
31. Gauch, S., Chaffee, J., Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent Systems **1** (2003) 219-234
32. Rabiner, L.R.: A tutorial on Hidden Markov Model and selected applications in speech recognition. Proceedings of the IEEE 77 (1989) 257-285

Appendix

Table 2. Comparison of the ontology-based focused Crawlers

Name	LSCrawler [16]	Courseware Watchdog crawler [15]	Crawler proposed by Ganesh [6]	THESUS crawler [8]
Domain	General	E-learning	General	General
Working Environment	General	Courseware Watchdog	General	THESUS
Special Functions	Computing and indexing the similarity values between URLs and topics.	Assigning weight to ontological concepts based on users' preference; weighting, ranking and clustering webpages based on the weighted concepts.	Weighting similarity values between URLs and ontological concepts, and between parent pages and children pages.	Linking URLs with ontological concepts.
Technologies Utilized	Ontology for similarity estimation; Porter Stemmer's algorithm for removing stop words.	Ontology for weighting, ranking and clustering webpages.	Ontology and combination importance, association and ordering metric for weighting similarity values between URLs and ontological concepts, and between parent pages and children pages.	Ontology and Boolean model for linking URLs with ontological concepts.
Evaluation Methods	Comparing recall rate with a full-text crawler.	Not provided.	Not provided.	Compared with a keyword-based clustering method on F-measure, rand statistics, etc.
Evaluation Results	Nearly 10% progress on recall rate, compared with the full-text crawler.	Not provided.	Not provided.	12% higher on F-measure, 5% advantage on rand statistics.
Comments/ Suggestions	Provide more evaluation details.	Provide evaluation details.	Provide evaluation details.	None.

Table 3. Comparison of the Metadata Abstraction Crawlers

Name	Vertical portal crawler [5]	CiteSeer crawler [7]
Domain	Legal	E-business
Working Environment	Vertical Portal	A niche search engine.
Special Functions	Collecting legal documents; abstracting metadata.	Parsing citations and abstracting metadata from downloaded documents.
Technologies Utilized	NB and MSVM for document classification; DC schema and tf-idf for metadata abstraction.	CiteSeer for parsing citations and abstracting metadata from downloaded documents; HMM for similarity estimation; SVM for metadata abstraction.
Evaluation Methods	Evaluating the classification accuracy values for NB and MSVM respectively.	Comparing SVM with HMM on accuracy.
Evaluation Results	82.5% for NB, 85.1% for MSVM.	SVM has better performance than HMM.
Comments/ Suggestions	Compare with other similar crawlers.	Evaluating with a bigger training set.

Table 4. Comparison of the Other Semantic Focused Crawlers (Part 1)

Name	Lokman crawler [2]	Crawler proposed by Liu et al. [12] [13]
Domain	Medical	General
Working Environment	MedicoPort	General
Special Functions	Fetching medical documents; estimating documents' similarity values to UMLS concepts; indexing unvisited URLs.	Predict relevant links based on users' preference.
Technologies Utilized	UMLS for ontology construction.	LSI for document clustering; HMM for similarity estimation; K-Nearest Neighborhood for children page clustering.
Evaluation Methods	Respectively comparing the harvest rate of the crawler in two algorithms with a Best-First crawler.	Comparing precision with a Best-First crawler.
Evaluation Results	Better than the Best-First crawler on overall harvest rate.	Significantly advantageous on precision than the Best-First crawler.
Comments/ Suggestions	Provide semantic functions for the fetched webpages.	Too many algorithms adopted could affect the overall efficiency.

Table 5. Comparison of the Other Semantic Focused Crawlers (Part 2)

Name	Web spider [3]	Digital library crawler [17]	BioCrawler [1]
Domain	General	Digital library	General
Working Environment	An agent-based semantic search engine.	CiteSeer	Semantic or non-semantic environment
Special Functions	Downloading all children pages; computing similarity values between downloaded webpages and predefined context.	Using metadata heuristics to retrieve missing publications in digital library.	Weighting the semantic strength of the obtained information based on its internal rules; sharing knowledge between crawlers; periodically revisiting websites to maintain its knowledge model.
Technologies Utilized	Algorithms for computing the semantic relevance between concepts and webpages.	Metadata heuristics for locating authors' homepages.	Not provided.
Evaluation Methods	Not provided.	Testing harvest level; comparing the crawler with a Breadth-First crawler in precision; comparing the crawler with a Hutch crawler in precision and speed.	Comparing the BioCrawler's energy with a dumb crawler during 30,000 websites visited and during 100 random restarts.
Evaluation Results	Not provided.	0.75 on harvest level; nearly 10% better than the Breadth-First crawler; superior than the Hutch crawler.	More knowledgeable than the dumb crawler, along with the increase of website visited.
Comments/ Suggestions	Downloading all children pages may affect the overall performance; provide evaluation details.	Its performance may vary in different venues.	Provide technical details.