Bedi P, Thukral A, Banati H *et al.* A multi-threaded semantic focused crawler. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(6): 1233–1242 Nov. 2012. DOI 10.1007/s11390-012-1299-8

A Multi-Threaded Semantic Focused Crawler

Punam Bedi[1], *Member*, *ACM*, *Senior Member*, *IEEE*, Anjali Thukral[2,*], Hema Banati[3], Abhishek Behl[1,**], and Varun Mendiratta[1,**]

[1] *Department of Computer Science, University of Delhi, Delhi-110007, India*

[2] *Department of Computer Science, Keshav Mahavidyalaya, University of Delhi, Delhi-110007, India*

[3] *Department of Computer Science, Dyal Singh College, University of Delhi, Delhi-110007, India*

E-mail: punambedi@ieee.org; athukral@cs.du.ac.in; hema.banati@gmail.com; {abhishek.behl, varun.mendiratta}@aricent.com

Received July 20, 2011; revised August 11, 2012.

Abstract The Web comprises of voluminous rich learning content. The volume of ever growing learning resources however leads to the problem of information overload. A large number of irrelevant search results generated from search engines based on keyword matching techniques further augment the problem. A learner in such a scenario needs semantically matched learning resources as the search results. Keeping in view the volume of content and significance of semantic knowledge, our paper proposes a multi-threaded semantic focused crawler (SFC) specially designed and implemented to crawl on the WWW for educational learning content. The proposed SFC utilizes domain ontology to expand a topic term and a set of seed URLs to initiate the crawl. The results obtained by multiple iterations of the crawl on various topics are shown and compared with the results obtained by executing an open source crawler on the similar dataset. The results are evaluated using Semantic Similarity, a vector space model based metric, and the harvest ratio.

Keywords eLearning, semantic focused crawler, semantically expanded term, ontology

1 Introduction

The Web is changing at a very fast pace, whether it be the content versatility or it be the technology that explores web content in meaningful and useful information. Fig.1 shows the pyramid of the Web evolution[1]. The divisions on the pyramid represent the volume of information in each version of the Web. The right side shows the technologies that have been used or expected to be used in future, during the evolving periods (left side of pyramid) of each Web version.

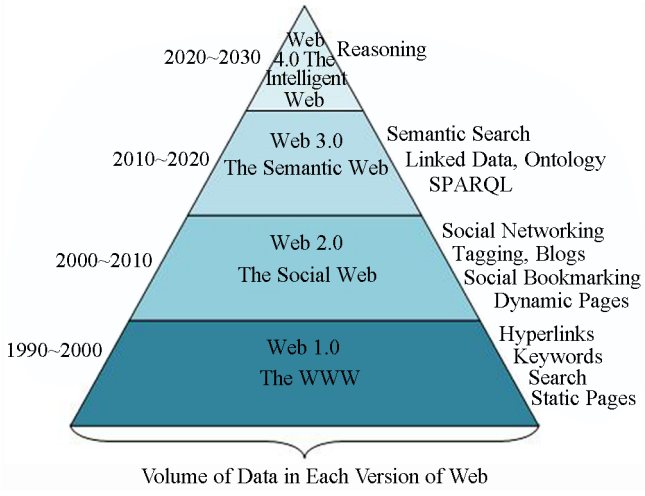

Fig.1. Different versions of the Web and associated technologies.

The World Wide Web (Web 1.0) which is primarily based on hyperlinks requires keywords, co-occurrence and page rank for searching relevant web pages. The relevance of web pages in this version of the Web is usually computed using hubs and authorities[2] and, keyword term frequency[3]. However, besides being simple and good computational techniques that search relevant web pages, the traditional search algorithms lack in searching semantically relevant web pages[4]. This means that the pages that contain synonyms, hypernyms or hyponyms for the keywords rarely get incorporated during the search. Further, Web 2.0, allows its users to interact or share their opinion through blogs, tagging, bookmarking sites, social networks, and so on, forming the social web altogether. Tagged web pages help in improving the search of relevant web pages[5] on the Web, but inclusion of semantic knowledge (ontology and axiom) along with tagged web resources makes the search more fruitful. In comparison to earlier Web versions, Semantic Web (Web 3.0) organizes

Regular Paper

*Corresponding Author

**MSc (Comp. Sc.) student at University of Delhi, India, at the time of the research of the paper

data in a different way. It stores all information in the form of linked data instead of hyperlinked web pages. It uses multiple ontologies/RDF (resource description framework)/RDFS (RDF schema)/XML (extensible markup language) linked together to form a Giant Global Graph[6] through which the required data can be extracted semantically. The intelligent web (Web 4.0) is believed to use reasoning-based recommendations built on the ontologies and logical axioms[1,7]. Although, Web 4.0 is termed as the next version of the semantic web, the distinguished features of Web 4.0 have already been incorporated as a part of the Semantic Web Architecture[8-9]. Hendler and Lee[10] designed the Web by incorporating all these features in their creation (the proposed web). Moreover, the future Web is believed to consist of the documents (all web pages linked through hyperlinks) as well as the data (the linked data in the form of ontologies and RDFs). However, the techniques and methods for searching and retrieving relevant content would connect these versions[11]. We believe that utilizing semantic data from Semantic Web technologies with the help of specially designed crawlers would benefit in searching relevant web pages from the WWW.

A web crawler in a generic or classic form is a program that traverses the Web through hyperlinks to index web pages in the local repositories[12] so as to provide better services to search engines and similar applications[13-14]. The focused crawlers instead perform a topical crawl[15] on the Web. They traverse topic specific hyperlinked web pages using various techniques to reach the topic relevant web pages. They serve many web-based applications, a few of which are discussed ahead. They are used by many search engines and web portals to build their web page repositories. They are useful to update topic relevant indexes and web portals where specific information is required to fulfill the community's information need in comparatively much lesser time. Dong and Hussain[16] have shown their use in industrial digital ecosystems for automatic service discovery, annotation and classification of information. In eLearning the crawlers are trained to collect learning content related to a specific topic for a learner as shown in this paper. The overall performance of a focused crawler mainly depends on the method of determining the priority of web pages to be crawled, which improves the harvest ratio (fraction of relevant web pages among total crawled web pages[8]) of a focused crawler. The priority computation usually includes methods to determine the relevance of web pages, and/or the path to reach relevant web pages. Therefore the major task during the focused crawl is to predict the ordering of web page visits. Some early designs of focused crawlers parsed anchor text to compute the relevance of web pages[17]. The web page relevance was also predicted by analysing the link structure and content similarity[18]. A similar research[19] calculated the link score based on average relevance score of parent web pages and division score (keywords related to the topic category) to determine the web page relevance. This was computed by taking term frequency of top 10 weighted common words from a set of seed pages which in turn was a set of common URLs retrieved by three search engines. Such approach, in some particular cases may yield URLs from an undesired domain, which consecutively results in wrong fetches. However retrieving search topic related keywords from a domain ontology eliminates the problem of selecting out-of-context keywords. Moreover computing semantic relevance over hyperlinked structures or PageRank algorithms[20] overcomes the problem of search engine optimization[21-23].

Several types of focused crawlers have surfaced in literature till date. On the basis of their design, focused crawlers can be categorized into two types, classic focused crawlers and learning-focused crawlers. Learning-focused crawler[24] applies a training set consisting of relevant and non-relevant web pages that governs (from learning) the selection of seed URLs and/or criteria to determine the relevance of a web page. On the other side, classic focused crawler works on some criteria[25] to form a path to reach relevant web pages. Based on these criteria, it has its two variants social semantic focused crawler[26] and semantic focused crawler. A semantic focused crawler utilizes domain knowledge to compute the relevance of web pages, whereas a social semantic focused crawler[25] uses bookmarked (tagged) web links on social web sites which may or may not use semantic knowledge to prioritize the sequence of web page traversal. The social semantic focused crawler computes page relevance based on the popularity of the web pages and manually assigned tags by user community. Although in comparison to the semantic focused crawler, the social semantic focused crawler retrieves relevant results without incurring the overhead of parsing the content of each web page, yet in addition it requires to pursue deep web search on social portals which makes the retrieval system dependent on the credibility of such sites. Also, only a small fraction of the Web which has been bookmarked by the user community is accessed by the crawler. Therefore this paper proposes SFC, a semantic focused crawler that searches relevant web resources in the WWW using the semantic knowledge related to the search topic.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 explains the framework and working of the proposed multi-threaded

semantic focused crawler. Experimental study and evaluation is discussed in Section 4. Section 5 concludes the paper.

2 Related Work

Existing work related to semantic focused crawlers mainly focuses on building them using ontology, but the way ontology is being utilized by these crawlers depends on the search motive. Thus, semantic focused crawlers can be categorized into two types, based on the search motive. One of them is specifically designed to search relevant ontologies in the WWW and the Semantic Web, usually used for "ontology search engines" such as Swoogle[27], OntoKhoj[28], OntoMetric[29], or AkTiveRank[30]. They crawl ontology repositories to gather linked data (in RDF, XML or OWL format) existing on the Web. Hence at the core level, they search ontologies and rank them according to the concept density within ontologies. The other type of the semantic focused crawlers searches relevant web pages (documents and not ontologies) from the Web by utilizing a pre-existing semantic knowledge to determine web page relevance (such types of semantic crawlers are discussed below). Thus, the former retrieves relevant ontologies while the latter retrieves semantically relevant web pages. Our paper focuses on the latter type of semantic focused crawlers. They are sometimes also referred as ontology-based focused crawlers.

The literature has a few reviews on ontology-based focused crawlers or semantic focused crawlers[31-32]. Ehrig *et al.*[33] computed relevance score by establishing entity reference using TF-IDF (term frequency-inverse document frequency) weights on natural language keywords and background knowledge compilation based on ontology. They refer ontology at each step to gather the relevance score which may make the computational time expensive. Moreover TF-IDF algorithm is usually applied on a large corpus to use it effectively[3,34]. The approach by Diligenti *et al.*[35] uses context graphs to find short paths that lead to relevant web pages. THESUS crawler[36] organizes web page collection based on incoming links of a web page and thereby clusters the web pages. The ontology-based Web crawler proposed by [37] computes similarity between web pages and ontological concepts by exploring association between parent page and children pages whereas courseware watchdog crawler[38] is built on the KAON system[39] which utilizes the user feedback to the retrieved web pages. However, both the papers have not discussed the evaluation details of their conceptual framework. LSCrawler[40], a general focused crawler, was built to index web pages by computing similarity between web pages and a given topic which shows the result comparisons with a full text crawler.

The proposed semantic focused crawler works differently from the above mentioned ontology-based crawlers as they are based on natural language semantics to crawl on the Web. The SFC proposed in this paper uses a specially designed domain ontology that consists of various educational concepts linked together semantically or conceptually[41] with an intent to search semantically relevant information on the Web. The unique feature of this crawler is that it uses dynamic semantic relevance (DSR) to prioritize the crawling list of the fetched web pages. The important thing to notice here is that the weights used to determine the semantic distance between two concepts are computed from the domain ontology, which in many research papers are assigned manually and stored in ontology for semantic computation. SFC intends to work for an e-mentoring application. It crawls the Web to retrieve semantically relevant learning content. These relevant resources are then augmented to ontology with appropriate links to its various concepts[42]. The knowledge base thus generated is consumed by the application that delivers the learning content to learners. Besides e-mentoring, this approach can also be applied to various other domains to retrieve semantically relevant web pages.

3 Proposed Multi-Threaded Semantic Focused Crawler

The proposed SFC is a focused crawler that uses multi-threading to crawl web pages relevant to the search topic. SFC utilizes the domain ontology to expand the topic. These domain ontologies are specifically designed for educational purpose to include maximum concepts that belong to a domain in a structured way. A brief introduction to domain ontology is given below followed by the SFC framework.

3.1 Domain Ontology

In information sciences, ontology has a status of resource, representing the conceptual model underlying a certain domain, describing it in a declarative way and thus separating it from procedural aspect[43]. As this paper deals with the learning content, the ontology, which is being utilized by SFC is therefore specially designed by linking various concepts that belong to different learning subjects such as "database", "computer networks", "computer organization". Various concepts under a domain are linked together by forming the relationships as in Concept Map[44] and Simple Knowledge Organization System (SKOS)[45]. The relationships such as `hasStatement, hasClause, isA, partOf, hasArithmeticOperators` have been derived from the Concept Map technology to analyse domain-

based relationships. The relationships derived from SKOS (such as `hasSuperconcept, hasSubconcept` and `isRelatedTo`) represent the generic relations such as broader, narrower and related concepts. The domain ontology thus built is therefore also termed as concept ontology. The concept or domain ontology is written in formal Web Ontology Language (OWL) to make the application semantic Web enabled. This ontology defines the structure of concepts under a domain, and therefore it is treated differently from the natural language ontology such as WordNet①. Significant information on computing semantic similarity measures that utilizes taxonomies can be found in [46-47]. The crawl is made for each concept in the domain ontology which is termed as search concept. The concept is expanded semantically[5] using the same ontology. The expanded list of a search topic includes alternate terminologies and abbreviations used to refer the search topic, in addition to all parent and child terms. The semantic distance from each expanded concept to the search concept is computed and stored in an array to be accessed by SFC threads. The proposed SFC helps to find potentially relevant web pages to annotate the concepts that exist in the ontology.

3.2 Semantic Focused Crawler Framework

The SFC framework is illustrated in Fig.2. It consists of domain ontology, priority queue, local database and the proposed multi-threaded semantic focused crawler. SFC runs multiple threads, where each thread picks up a top priority URL from the priority queue which is a web page with highest dynamic semantic relevance (DSR) (explained later in detail). The threads independently parse the web page and extract all hyperlinks on that page to put again on the queue. These hyperlinks are then fetched again and parsed one by one to compute DSR. The priority queue thus, maintains the order of URLs to be parsed by SFC threads. During the crawl each thread also checks for already visited URLs to avoid cycles. For this purpose a separate temporary queue is maintained (not shown in the framework) which stores all visited URLs. All URLs of potentially relevant web pages fetched during the crawling process are stored in the local database, to be later consumed by other applications. Each thread of SFC carries out the crawl process in two parts, which are explained in detail below. Algorithm 1 presents a summary of the complete crawl procedure used by the proposed SFC.

Algorithm 1. Semantic Focused Crawler

${}_pQ$: priority queue containing URLs and their dynamic semantic relevance, DSR_{P_i}

${}_gLinks$: queue containing traversed URLs during the crawl to avoid cycles, thus it checks for duplicate traversals

${}_eT$: expanded topic list consists of related terms (concepts) and their semantic distances to the topic from the ontology

Thus, ${}_eT = \{(c_0, 0), (c_1, d_1), (c_2, d_2), \ldots, (c_n, d_n)\}$, where $n > 0$, $T = \{t_0, t_1, t_2, \ldots, t_m\}$, where $m > 0$, t_0 is the topic for the focused crawl, t_i: semantically related to concept c_j, d_i: semantic distance.

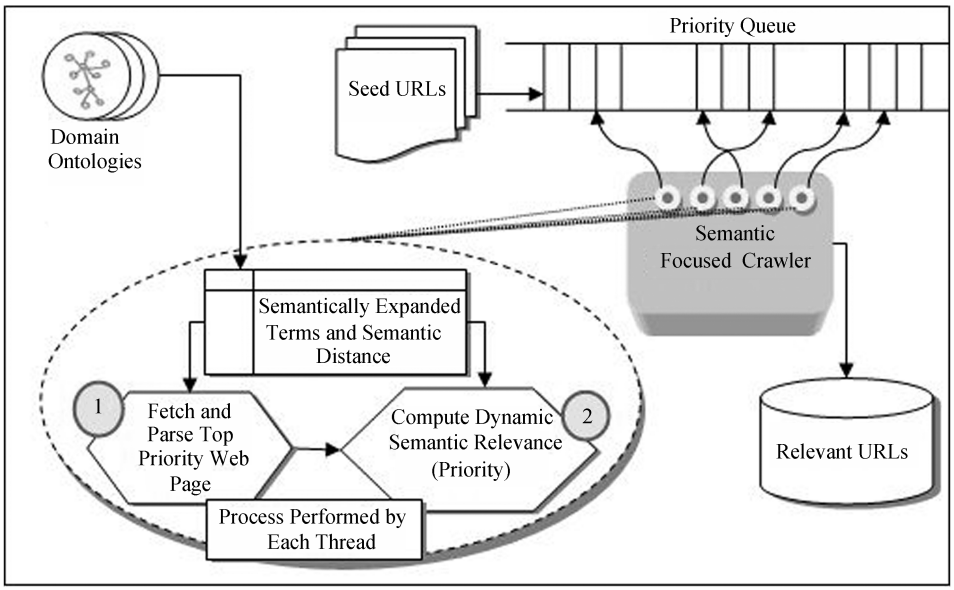

Fig.2. Framework of proposed SFC.

①http://wordnet.princeton.edu

```
1.  Initialize pQ with seed URLs
2.  Repeat till (!pQ.empty() ||fetch_cnt ⩽ Limit) {
3.    web_page.url = pQ.top.getUrl();
      //Get most relevant single URL from priority queue
4.    Fetch and parse web_page.url;
5.    web_page.urls = extract URLs (hyperlinks) from
      web_page.url;   //List of URLs
6.    For each web_page.urls {
7.      already_exist = Check web_page.urls[i] in gLinks;
        //Check for duplicates
8.      If (!already_exist) {
9.        Enqueue web_page.urls[i] in gLinks;
10.       Fetch and parse web_page.urls[i];
11.       Compute DSR_Pi of web_page.urls[i];
12.       Enqueue (web_page.urls[i], DSR_Pi) in pQ;
13.       Store (web_page.urls[i], DSR_Pi) in local
          database;
14.     }   //end of If
15.   }   //end of For each
16. } //   end of Repeat
```

3.2.1 Fetch and Parse a Web Page

The priority queue is initiated with the seed URLs, which can be fetched from a search engine. Dynamic semantic relevance (DSR) of these resources is computed and then enqueued to the priority queue. A web page with the top priority URL from the priority queue is fetched from the Web (shown as number "1" in Fig.2). The web page source is then parsed to extract the URLs (hyperlinks) and tokenized to determine the frequency of each concept in the expanded concept list. This extracted data is then consumed by the next process to determine DSR.

3.2.2 Compute Dynamic Semantic Relevance

The dynamically computed semantic relevance DSR_{P_i} of each web page $\boldsymbol{P}_i$, is a distinguished feature of this SFC. DSR is computed after a web page is parsed during the crawl process. Thereafter, the web page is placed onto the priority queue along with its computed DSR. In the next spanning iteration, the thread picks up a web page with highest DSR from the priority queue so as to reach all those web pages which are linked to the parent web page. This is based on the assumption that a web page, which is considered highly relevant would contain hyperlinks to more relevant web pages, therefore the hyperlinks on this web page should be crawled first. In this way, the web pages that are more relevant to a search topic would get priority to be crawled first over the less relevant web pages.

Dynamic semantic relevance DSR_{P_i} of a web page $\boldsymbol{P}_i$ to a topic t_0 is computed by following steps.

Step 1. Topic t_0 for focused crawling is expanded from the domain ontology D_i where $t_0 \in D_i$. The topic is expanded by including all parent nodes and a few levels[②] of child nodes from the ontology. To avoid ontology access during the crawl, a structure comprising each associated concept (term) and its semantic distance (explained below) is stored in a temporary memory. This reduces the time spent on frequent access and traversal of the ontology. The domain ontology for the purpose is created as a semantic graph, consisting of various concepts from the education and learning perspective (as explained above).

Step 2. The semantic distance (SD) between a topic and all other concepts in ontology is computed using the following formula,

$$SD = |d|_{C_i,C_j}. \tag{1}$$

Here, $|d|$ is the number of edges or links between any two concepts, C_i and C_j. The concepts in ontology are the terms that belong to a learning domain (or a particular subject).

Step 3. Semantic relevance between two concepts belonging to the domain ontology is

$$SR_{C_i,C_j} = \frac{1}{|d|_{C_i,C_j} + 1}. \tag{2}$$

Thus, the semantic relevance is inversely proportional to the distance between any two concepts in ontology.

Step 4. Dynamic semantic relevance, DSR_{P_i} of a web page, $\boldsymbol{P}_i$, with respect to a topic (C_i) is calculated by summing up the product of the frequency of each term (exist in the expanded list) in the web page and its semantic relevance SR_{C_i,C_j} (see (2)). This is formalized as following.

$$DSR_{P_i} = \sum_{j=1}^{n} (f_{C_j} \times SR_{C_i,C_j}). \tag{3}$$

Here, n is the total terms (concepts: C_j) in the expanded topic list and f_{C_j} is the frequency of a concept C_j in web page $\boldsymbol{P}_i$.

3.3 Evaluation Metric

Semantic Similarity Model Used by SFC. Semantic similarity between web pages retrieved by the crawlers and the expanded topic list have been used to evaluate

②This particular case takes concepts up to the 4th level in ontology, although this may vary according to the depth of content required on the topic.

the two crawlers for the experimental purpose. The semantic similarity measure is based on the Vector Space Model (VSM)[3] where the term weights are computed from the domain ontology to generate a topic vector. Similarly, the semantically related terms from a web page produces a web page vector. These vectors determine the *cosine* θ similarity. The vector lengths③ of a topic vector and a web page vector are computed using following methods.

The topic vector length $|\boldsymbol{T}|$ is computed using the semantic distance between the topic term and each term from its expanded list as the following function:

$$|\boldsymbol{T}| = \sqrt{\sum_{j} Wt_{t_0,t_j}^2}. \tag{4}$$

Here, Wt_{t_0,t_j} is the term weight computed from the ontology using (5).

$$Wt_{t_0,t_j} = \frac{SR_{t_0,t_j}}{\sum_{j=0}^{m} SR_{t_0,t_j}}. \tag{5}$$

Here, SR is the semantic relevance computed between $t_0 \rightarrow t_j$, using the domain ontology (from (2)).

Similarly, a web page vector length is computed for each web page ($\boldsymbol{P}_i$), using the following function:

$$|\boldsymbol{P}_i| = \sqrt{\sum_{k} Wt_{p_k,P_i}^2}, \tag{6}$$

where Wt_{p_k,P_i} is the weight computed for each term $p_k \in T$ that exists in the web page $\boldsymbol{P}_i$ so that,

$$Wt_{p_k,\boldsymbol{P}_i} = \frac{SR_{t_0,p_k} \times f_{p_k}}{\sum_{k=1}^{m}(SR_{t_0,p_k} \times f_{p_k})}. \tag{7}$$

The normalized cosine similarity between t_0 and a web page P_i is defined as:

$$Cosine\theta_{t_0,P_i} = \frac{\boldsymbol{T} \cdot \boldsymbol{P}_i}{|\boldsymbol{T}| \times |\boldsymbol{P}_i|}, \tag{8}$$

where, $\boldsymbol{T} \cdot \boldsymbol{P}_i$ is the dot product computed as,

$$\boldsymbol{T} \cdot \boldsymbol{P}_i = \sum_{t_j=p_k} (Wt_{t_0,t_j} \times Wt_{p_k,P_i}). \tag{9}$$

Therefore, using (4), (6), (8) and (9), we get,

$$Cosine\theta_{t_0,P_i} = \frac{\sum_{t_j=p_k}(Wt_{t_0,t_j} \times Wt_{p_k,P_i})}{\sqrt{\sum_{t_j} Wt_{t_0,t_j}^2} \times \sqrt{\sum_{k} Wt_{p_k,P_i}^2}}. \tag{10}$$

For the most relevant web page, the value of similarity measure ($Cosine\theta$) is the maximum and it is 0 for irrelevant web pages. Each web page is evaluated for its similarity with the topic using above method.

Harvest Ratio. Harvest ratio (hr)[15,24-25,48-49] is defined as the fraction of web pages crawled that satisfy the crawling target (relevant pages) $\#r$ among the crawled pages $\#p$. Thus $hr = \#r/\#p$, $hr \in [0, 1]$. It determines the efficiency of a crawler to retrieve larger number of relevant web pages among the total retrieved. The relevance of the retrieved web pages is determined using semantic similarity model described above.

4 Experimental Study and Evaluation

4.1 Experimental Setup

The experiment was conducted on various topics from different domains. The crawlers were executed on Intel Core 2 Duo processor, 2.4 GHz, 2 GB RAM, 32-bit OS. Several iterations of the crawling process were performed on every set of all topics to ensure the quality of results.

An open source multi-threaded java crawler called crawler4j④ was used as a baseline crawler to compare the results of SFC, as the open source for any type of semantic focused crawlers were not available. It was customized to incorporate expanded list of the topics and seed URLs by overriding the methods *shouldVisit*(WebURL *url*) and *visit*(Page *page*). Hence, for this experiment the Crawler4j will be referred as Classic Focused Crawler (FC) or "Classic FC" to denote the customized crawler. The results of both crawlers were compared by taking the same set of seed URLs, topic and number of threads.

The difference between the designs of two crawlers lies in the priority computation based on the semantic relevance. Classic FC is a baseline focused crawler with first come first serve (FCFS) priority, whereas the proposed crawler SFC uses dynamic semantic relevance (DSR) to prioritize the visiting sequence of web pages. All relevant URLs retrieved from both of the crawlers are stored separately in MySql database during the crawl.

The semantic relevance of the fetched web pages is computed using the semantic similarity model explained in Subsection 3.3. An adjustable threshold is used to filter relevant web pages. The threshold may include one of the values from the range ("> 0", "$\geqslant 0.2$", "$\geqslant 0.4$", "$\geqslant 0.6$"). The value of the threshold is adjusted manually so that nearly 50% of top relevant web

③A vector consists of two components, a direction and a magnitude (length). In information retrieval, direction has no meaning therefore, only length is used for computational purpose.

④http://code.google.com/p/crawler4j/

pages get incorporated in the experimental evaluation. This allows evaluating crawlers more rigorously, because the high threshold implies high value of relevance for a web page. The crawled results of both crawlers are evaluated using semantic similarity measure based on Vector Space Model. The results are compared and analyzed using harvest ratio.

4.2 Results

Since it is not possible to show the evaluation on all topics, the evaluation on three topics is discussed in detail. They are purposefully chosen from different domains. The topics are "dml", "transmission_media" and "integrated_circuit" from "database", "computer networks" and 'computer organization' domains respectively. Fig.3 shows the number of resources retrieved by SFC and the customized Crawler4j, called Classic FC. The input datasets for both crawlers are the same. The comparison is made for unique number of resources retrieved, and relevant resources with the threshold (semantic relevance value) greater than 0, and greater than 0.2. It shows that the Classic FC though retrieved larger number of unique web resources, larger number of relevant web resources were retrieved by SFC. The analysis infers the efficiency of SFC which retrieved 88% of relevant resources while Classic FC retrieved only 31%. The graphs in Figs. 4, 5 and 6 represents the harvest ratio (explained in Subsection 3.3) of the SFC and Classic FC crawler with three different topics respectively.

Fig.4 shows harvest ratio of the crawlers when crawled on the topic "dml". The relevant resources are measured at threshold "$\geqslant 0.4$". The initial high ratio, in the graph is seen due to the seed URLs, most of which are relevant. However all the links on those pages may not be relevant, as a result the ratio falls down for a short period. Later the ratio picks up and becomes steady because of more relevant crawls. It shows that SFC crawled approximately. 40% relevant web

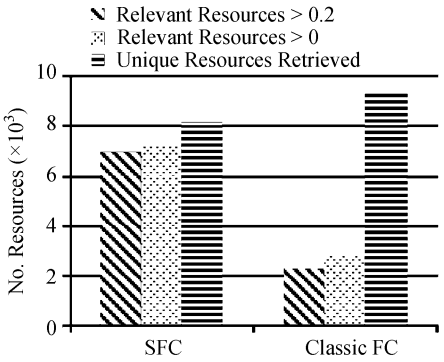

Fig.3. Number of resources retrieved by the two crawlers on same topic and same set of the seed URLs.

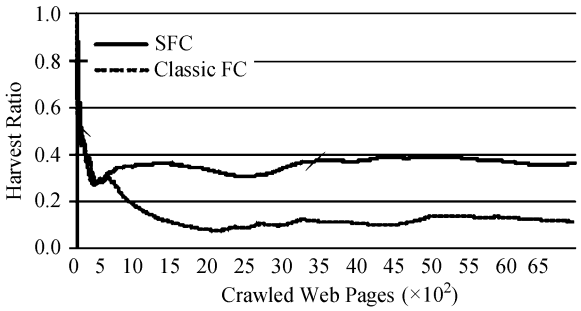

Fig.4. Harvest ratio for the topic "dml".

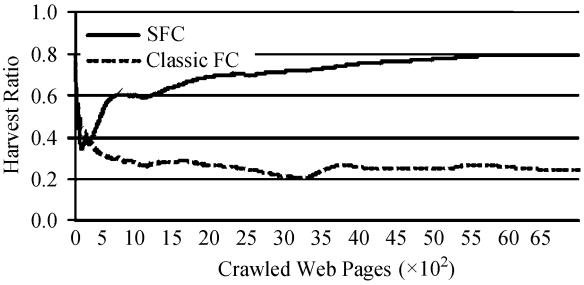

Fig.5. Harvest ratio for the topic "transmission media".

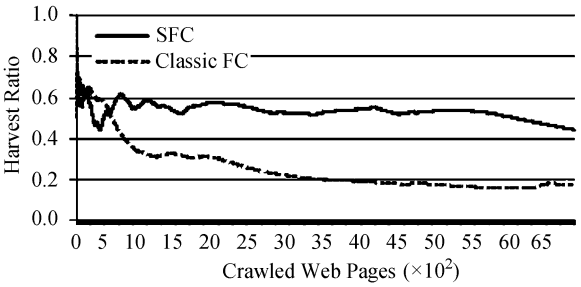

Fig.6. Harvest ratio for the topic "integrated circuit".

resources which have semantic relevance more than 0.4; whereas Classic FC retrieved approximately 15% relevant resources.

Fig.5 shows the harvest ratio of SFC and Classic FC for the topic "transmission media" at the threshold "$\geqslant 0.2$". It shows contrasting results between SFC and the baseline crawler, with the harvest ratio of SFC as good as 0.8.

A harvest ratio comparison of SFC and Classic FC for the topic "integrated circuit" at the threshold "$\geqslant 0.2$" is shown in Fig.6. It shows retrieval of approximately 60% relevant web resources by SFC and approximate 20% by Classic FC, where web pages with semantic relevance greater than or equal to 0.2 have been considered as relevant resources. SFC has shown an improvement over Classic FC for retrieving semantically relevant web resources from the Web, of 58%, 55% and 54% when crawled on the topics, "dml", "transmission media" and "integrated circuit" respectively. It is

apparent from the overall results that the use of dynamic semantic relevance during crawl (used by SFC) retrieves larger number of relevant web pages than the simple FCFS-based crawl (used by Classic FC).

5 Conclusions

The design and implementation of a multi-threaded semantic focused crawler (SFC) was presented in this paper. The crawler was used to fetch semantically relevant web pages from the Web on given topics. SFC uses dynamic semantic relevance (DSR) to prioritize the web pages to be crawled further. DSR is computed during the crawl for each web page, based on the expanded list of the topic and the semantic distances among various semantically linked concepts from the domain ontology. Domain ontology is constructed manually on a few learning subjects, to include most of the related concepts which are linked based on their semantic relations. The potentially relevant web pages found by the SFC are stored in a local database.

SFC was evaluated using seed URLs from a search engine on various topics, among which crawl results of three topics belonging to different domains were discussed in detail. An open source multi-threaded focused crawler, Crawler4j was customized to crawl web pages on FCFS basis, hence was named as Classic FC for experiment purpose in this paper. The results from both crawlers were compared and evaluated for the same dataset and the evaluation model. SFC which uses domain ontology and DSR for crawling web pages, performed better (showed improvement of 55% on an average) for all given topics over Classic FC crawler.

In future, the crawler will be extended to incorporate social relevance in addition to the semantic relevance. The crawler will also be implemented as a software agent and would be capable of communicating and cooperating with other agents to share knowledge bases for acquiring semantic knowledge and to provide services to other agents.

Acknowledgement Authors would like to acknowledge the Editor-in-Chief and the anonymous reviewers for their valuable suggestions to improve this paper.

References

[1] Spivack N. Web evolution. http://www.slideshare.net/novaspivack/web-evolution-nova-spivack-twine, June 2011.

[2] Kleinberg J, Lawrence S. The structure of the Web. *Science*, 2001, 294(5548): 1849-1850.

[3] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 1988, 24(5): 513-523.

[4] Navigli R, Velardi P. An analysis of ontology-based query expansion strategies. In *Proc. Workshop on Adaptive Text Extraction and Mining*, Sept. 2003, pp.42-49.

[5] Bedi P, Banati H, Thukral A. Social semantic retrieval and ranking of eResources. In *Proc. the 2nd Int. Conference on Advances in Recent Technologies in Communication and Computing*, Oct. 2010, pp.343-347.

[6] Berners-Lee T. Giant global graph. http://dig.csail.mit.edu/breadcrumbs/node/215, May 2011.

[7] Farber D. From semantic Web (3.0) to the WebOS (4.0). http://www.zdnet.com/blog/btl/from-semantic-web-30-to-the-webos-40/4499, May 2011.

[8] Berners-Lee T, Hendler J, Lassila O. The semantic web. *Scientific American*, 2001 284(3): 34-43.

[9] Bedi P, Banati H, Thukral A. Use of ontology for reusing web repositories for eLearning. In *Technological Developments in Networking, Education and Automation*, Elleithy K *et al.* (eds.), New York, USA: Springer, 2010, pp.97-101.

[10] Hendler J, Berners-Lee T. From the semantic web to social machines: A research challenge for AI on the World Wide Web. *Artificial Intelligence*, 2010, 174(2): 156-161.

[11] Berners-Lee T. Semantic Web and linked data. http://www.w3.org/2009/Talks/0120-campus-party-tbl/, June 2011.

[12] Pant G, Srinivasan P, Menczer F. Crawling the web. In *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*, Levene M, Poulovassilis A (eds.), Springer-Verlag, 2004, pp.153-178.

[13] Castillo C. Effective Web crawling [Ph.D. Thesis]. Dept. of Computer Science, University of Chile, November 2004.

[14] Bidoki A M Z, Salehie M, Azadnia M. Analysis of priority and partitioning effects on web crawling performance. In *Proc. the Intelligent Information Processing and Web Mining Conference*, May 2004, pp.287-296.

[15] Chakrabarti S, van den Berg M, Dom B. Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks*, 1999, 31(11-16): 1623-1640.

[16] Dong H, Hussain F K. Focused crawling for automatic service discovery, annotation and classification in industrial digital ecosystems. *IEEE Transactions on Industrial Electronics*, 2011, 58(6): 2106-2116.

[17] Craswell N, Hawking D, Robertson S. Effective site finding using link anchor information. In *Proc. the 24th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, Sept. 2001, pp.250-257.

[18] Jamali M, Sayyadi H, Hariri B B, Abolhassani H. A method for focused crawling using combination of link structure and content similarity. In *Proc. IEEE/WIC/ACM Int. Conference on Web Intelligence*, Dec. 2006, pp.753-756.

[19] Hati D, Kumar A. An approach for identifying URLs based on division score and link score in focused crawler. *Int. Journal of Computer Application*, 2010, 2(3): 48-53.

[20] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the Web. In *Proc. the 7th Int. WWW Conference*, April 1998, pp.161-172.

[21] Callen B. Search Engine Optimization Made Easy. http://www.easywebtutorials.com / ebooks / SEO - MadeEasy . pdf, June 2011.

[22] The Bivings group. SEO basics. http://www.knightdigitalmediacenter.org/images/uploads/leadership/SEO%20Basics.pdf, June 2011.

[23] Google. Search engine optimization starter guide. http://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf, June 2011.

[24] Batsakis S, Petrakis E G M, Milios E. Improving the performance of focused web crawlers. *Data & Knowledge Engineering*, 2009, 68(10): 1001-1013.

[25] Thukral A, Mendiratta V, Behl A, Banati H, Bedi P. FCHC: A social semantic focused crawler. In *Proc. Int. Conf.*

Advances in Computing and Communications, July 2011, pp.273-283.

[26] Thukral A, Banati H, Bedi P. Ranking tagged resources using social semantic relevance. *Information Retrieval Research*, 2011, 1(3): 15-34.

[27] Ding L, Finin T, Joshi A *et al.* Swoogle: A search and metadata engine for the semantic web. In *Proc. the 13th ACM Conf. Information and Knowledge Management*, Nov. 2004, pp.652-659.

[28] Patel C, Supekar K, Lee Y, Park E K. OntoKhoj: A semantic web portal for ontology searching, ranking and classification. In *Proc. the 5th ACM Int. Workshop on Web Information and Data Management*, Nov. 2003, pp.58-61.

[29] Lozano-Tello A, Gómez-Pérez A. ONTOMETRIC: A method to choose the appropriate ontology. *Journal of Database Management*, 2004, 15(2): 1-18.

[30] Alani H, Brewster C, Shadbolt N. Ranking ontologies with AKTiveRank. In *Proc. the 5th Int. Conf. Semantic Web*, Nov. 2006, pp.1-15.

[31] Dong H, Hussain F K, Chang E. A survey in semantic web technologies-inspired focused crawlers. In *Proc. the 3rd Int. Conf. Digital Information Management*, Nov. 2008. pp.934-936.

[32] Dong H, Hussain F K, Chang E. State of the art in semantic focused crawlers. In *Proc. Int. Conference on Computational Science and its Applications*, June 29-July 1, 2009, Part 2, pp.910-924.

[33] Ehrig M, Maedche A. Ontology-focused crawling of Web documents. In *ACM Symposium on Applied Computing*, March 2003, pp.1174-1178.

[34] Garcia E. The classical vector space model: Description, advantages and limitations of the classic vector space model. http://www.miislita.com/term-vector/term-vector-3.html, Oct. 2010.

[35] Diligenti M, Coetzee F, Lawrence S, Giles C, Gori M. Focused crawling using context graphs. In *Proc. the 26th Int. Conference on Very Large Data Bases*, Sept. 2000, pp.527-534.

[36] Halkidi M, Nguyen B, Varlamis I, Vazirgiannis M. THESUS: Organizing Web document collection based on link semantics. *Journal on Very Large Data Bases*, 2003, 12(4): 1-13.

[37] Ganesh S, Jayaraj M, Kalyan V *et al.* Ontology-based web crawler. In *Proc. Int. Conf. Information Technology: Coding and Computing*, April 2004, 2: 337-341.

[38] Tane J, Schmitz C, Stumme G. Semantic resource management for the web: An e-learning application. In *Proc. the 13th Int. World Wide Web Conference on Alternate Track Papers & Posters*, May 2004, pp.1-10.

[39] Maedche A, Staab S. Ontology learning. In *Handbook on Ontologies*, Staab S, Studer R (eds.), Springer-Germany, 2004.

[40] Yuvarani M, Iyengar N Ch S N, Kannan A. LSCrawler: A framework for an enhanced focused web crawler based on link semantics. In *Proc. Int. Conference on Web Intelligence*, Dec. 2006, pp.794-800.

[41] Thukral A, Bedi P, Banati H. Architecture to organize social semantic relevant web resources in a knowledgebase. *Int. Journal of e-Education, e-Business, e-Management and e-Learning*, 2011, 1(1): 45-51.

[42] Thukral A, Bedi P, Banati H. Automatic organization of web resources in ontologies for learning purpose. In *Proc. the 2nd Int. Conference on e-Education, e-Business, e-Management and E-Learning*, Jan. 2011, pp.38-44.

[43] Cimiano P. Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer Heidelberg, 2006.

[44] Novak J D. Learning, creating, and using knowledge: Concept maps as facilitative tools in schools and corporations. *Journal of e-Learning and Knowledge Society*, 2010, 6(3): 21-30.

[45] Isaac A, Summers E. SKOS: Simple knowledge organization system primer. http://www.w3.org/TR/skos-primer, Feb. 2011.

[46] Hliaoutakis A, Varelas G, Voutsakis E *et al.* Information retrieval by semantic similarity. *Int. Journal on Semantic Web and Information Systems*, 2006, 3(3): 55-73.

[47] Dong H, Hussain F K, Chang E. A context-aware semantic similarity model for ontology environments. *Concurrency and Computation: Practice and Experience*, 2010, 23(5): 505-524.

[48] Menczer F, Pant G, Ruiz M E, Srinivasan P. Evaluating topic-driven web crawlers. In *Proc. the 24th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, Sept. 2001, pp.241-249.

[49] Zheng H T, Kang B Y, Kim H G. Learnable focused crawling based on ontology. In *Proc. the 4th AIRS*, Jan. 2008, pp.264-275.

Punam Bedi received her M.Tech. degree in computer science from IIT Delhi, India in 1986 and her Ph.D. degree in computer science from the Department of Computer Science, University of Delhi, India in 1999. She is an associate professor in the Department of Computer Science, University of Delhi. She has about 25 years of teaching and research experience and has published about 140 papers in national/international journals/conferences. Dr. Bedi is a member of AAAI, ACM, senior member of IEEE, and life member of Computer Society of India. Her research interests include web intelligence, soft computing, semantic Web, multi-agent systems, intelligent information systems, intelligent software engineering, intelligent user interfaces, requirement engineering, human-computer interaction (HCI), trust, information retrieval and personalization.

Anjali Thukral is an assistant professor in the Department of Computer Science at Keshav Mahavidyalaya, University of Delhi, India. She is also a research scholar and her research interests include information retrieval, knowledge representation, ontology, eLearning, semantic web and multi-agent systems.

Hema Banati is working as an associate professor in the Department of Computer Science, Dyal Singh College, University of Delhi, India. She received her Ph.D. degree in computer science from the Department of Computer Science, University of Delhi in 2006. She has many national and international publications to her credit. Over the past decade she has been pursuing research in the areas of Web engineering, software engineering, human-computer interaction, multi-agent systems, eCommerce and eLearning.

Abhishek Behl got his MSc degree in computer science in 2011 from University of Delhi, India. He got his B.Sc. in computer science (Hons) from University of Delhi in 2009. He is currently working as a software engineer at one of the leading software development firms. His research interest includes artificial intelligence, development of Web-based and mobile-based applications.

Varun Mendiratta got his MSc degree in computer science in 2011 from University of Delhi, India. He got his B.Sc. degree in computer science (Hons) from University of Delhi in 2009. He is presently working as a software engineer at one of the leading software development firms. His research interest includes Web resources retrieval, development of Web-based and mobile-based applications.

Appendix

Table A1. List of Seed URLs for Different Search Topics and the Computed Semantic Similarity

Search Topic	Seed URL	Similarity
dml	http://en.wikipedia.org/wiki/Data_Manipulation_Language	0.756 298
	http://www.tomjewett.com/dbdesign/dbdesign.php?page=ddldml.php	0.938 018
	http://www.geekinterview.com/question_details/12782	0.896 983
	http://www.orafaq.com/faq/what_are_the_difference_between_ddl_dml_and_dcl_commands	0.918 717
	http://www.dmlgroup.in/mdsdesk.html	0.798 010
	http://www.dml.co.in	0
	http://www.directmylink.com	0
	http://dmlbuild.sourceforge.net	0.798 010
	http://en.wikipedia.org/w/index.php?title=Data_Manipulation_Language&action=edit	0.779 050
transmission_media	http://en.wikipedia.org/wiki/Transmission_medium	0.619 575
	http://www.webopedia.com/TERM/T/transmission_media.html	0.543 026
	http://www.techbooksforfree.com/intro_to_data_com/page37.html	0
	http://www.wiziq.com/tutorial/27574-Transmission-Media-for-Networking	0.395 923
	http://www.rigacci.org/docs/biblio/online/intro_to_networking/c1179.htm	0.513 850
	http://elearn.main.nvsu.edu.ph/ebooks/networking.essentials/5a671dd.htm	0.581 426
	http://www.geekinterview.com/question_details/79634	0.492 592
	http://penguin.dcs.bbk.ac.uk/academic/networks/physical-layer/media/index.php	0.395 875
	http://www.transmissionmedia.com/	0
integrated_circuit	http://www.tradeindia.com/suppliers/integrated-circuits.html	0.569 178
	http://nobelprize.org/educational/physics/integrated_circuit/history/	0.624 629
	http://integratedcircuits.tradeindia.com/	0.514 033
	http://schools-wikipedia.org/wp/i/Integrated_circuit.htm	0.732 019
	http://products.jimtrade.com/6/integrated_circuits.htm	0.369 484
	http://simple.wikipedia.org/wiki/Integrated_circuit	0.503 048
	http://dir.indiamart.com/impcat/integrated-circuits.html	0.424 106
	http://www.payscale.com/research/IN/Industry=Integrated_Circuits/Salary	0
	http://www.wikinfo.org/index.php/Integrated_circuit	0.779 316
	http://my.indiamart.com/cgi/eto-alerts-new.mp?modid=MY	0