What's There and What's Not? Focused Crawling for Missing Documents in Digital Libraries

Ziming Zhuang
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA
zzhuang@ist.psu.edu

Rohit Wagle
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA
rohitsw@psu.edu

C. Lee Giles
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA
giles@ist.psu.edu

ABSTRACT

Some large scale topical digital libraries, such as CiteSeer, harvest online academic documents by crawling open-access archives, university and author homepages, and authors' self-submissions. While these approaches have so far built reasonable size libraries, they can suffer from having only a portion of the documents from specific publishing venues. We propose to use alternative online resources and techniques that maximally exploit other resources to build the complete document collection of any given publication venue.

We investigate the feasibility of using publication metadata to guide the crawler towards authors' homepages to harvest what is missing from a digital library collection. We collect a real-world dataset from two Computer Science publishing venues, involving a total of 593 unique authors over a time frame of 1998 to 2004. We then identify the missing papers that are not indexed by CiteSeer. Using a fully automatic heuristic-based system that has the capability of locating authors' homepages and then using focused crawling to download the desired papers, we demonstrate that it is practical to harvest using a focused crawler academic papers that are missing from our digital library. Our harvester achieves a performance with an average recall level of 0.82 overall and 0.75 for those missing documents. Evaluation of the crawler's performance based on the harvest rate shows definite advantages over other crawling approaches and consistently outperforms a defined baseline crawler on a number of measures.

Categories and Subject Descriptors

H.3.7 [**Information Systems**]: Information Storage and Retrieval – *Digital Libraries*

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords: Digital libraries, focused crawler, CiteSeer, DBLP, ACM, harvesting.

JCDL'05, June 7–11, 2005, Denver, Colorado, USA

1. INTRODUCTION

Digital libraries that are based on active crawling methods such as CiteSeer often have missing documents in collections of archived publications, such as ACM and IEEE. How do such digital libraries find and obtain those missing? We propose using external resources of publication metadata and focused crawlers to search the Web for those missing.

The basic concept of a focused crawler (also known as a topical crawlers) [1], is based on a crawling strategy that relevant Web pages contain more relevant links, and these relevant links should be explored first. Initially, the measure of relevancy was based on keywords matching; connectivity-based metrics were later introduced [2]. In [3] the concept of a focused crawler was formally introduced: *a crawler that seeks, acquires, indexes, and maintains pages on a specific set of topics that represent a relatively narrow segment of the Web*.

Today, focused crawling techniques have become more important for building specialty and niche (vertical) search engines While both the sheer volume of the Web and its highly dynamic content increasingly challenge the task of document collection, digital libraries based on crawling benefit from focused crawlers since they can quickly harvest a high-quality subset of the relevant online documents.

Current approaches to harvesting online academic documents normally consist of focused crawling of open-access archives, author and institution web sites and directories of authors' self-submissions. A random sample of 150 journals and conferences in Computer Science show that less than 10% have websites that are open to crawlers. Many of the top publishing venues that have their documents electronically available to subscribers such as the ACM Digital Library, the IEEE Digital, Library or the Springer-Verlag Digital Library, normally use access permission techniques and robots.txt to ban crawlers. A recent study indicates that CiteSeer indexes 425, 000 unique research documents related to Computer Science, DBLP contains 500,464 records and there are 141,345 records in the Association for Computing Machinery (ACM) Digital Library and 825,826 records in the more comprehensive ACM Guide [4]. The study also shows that in CiteSeer there is an overlapping portion of 86, 467 documents (20.2% of CiteSeer's total archive) comprising 17.3% of the Digital Bibliography & Library Project (DBLP) archive.

This research investigates alternative online resources and focused crawling techniques to build a complete document

collection for any given publication venue. We propose to answer the following:

Q1 - *What are the best focused crawling techniques to maximally exploit online resources, in order to harvest the desired papers effectively and efficiently?*

Q2 – *Is it effective to use authors' homepages as alternative online resources to find the missing documents?*

Q3 – *How can the above methods be automated to effectively obtain missing documents?*

The rest of the paper is organized as follows. In section 2 we present a review of related work. In Section 3 we cover in much detail the design rationale of the system. In Section 4 we describe how we collect data and perform the evaluation, and present the results with discussion. Finally, we conclude the paper with future work proposed in Section 5.

2. RELATED WORK

The focused crawling literature shows that much has been focused on enhancing the dynamic performance, scalability, effectiveness, and efficiency of the crawler, namely, harvesting higher-quality documents in a shorter period of time.

Breadth-first searching is probably the simplest strategy for crawling, i.e. traversing the Web in a way that a directed graph is traveled using a breadth-first search algorithm. Interestingly, a breadth-first crawler is found to be capable of yielding high-quality documents at an early stage of the crawl [5]. Although more sophisticated crawlers tend to retrieve even higher quality pages than their breadth-first counterparts, they are usually computationally more expensive. In our study, we use a multi-threaded breadth-first crawler as a baseline to compare to our own crawling method.

Best-first crawling attempts to direct the crawler towards the *best* (i.e. most relevant in terms of topic relevance) documents. Different heuristics, such as link-based criteria, lexical similarity measures, contextual knowledge, and fine-tuned combinations of such have been explored in a number of studies over the years. In [2], the authors find that PageRank [6] can yield the best performance when ordering seed URLs. However, a more recent study [7] shows that PageRank metrics may just be too general in context without regard to the specific target topic. An updated version of PageRank algorithm which reflects the importance with respect to a particular topic has been proposed [8].

In [3], a Bayesian classifier is used to estimate the probability that a page belongs to the target topic, in a way that a node belongs to a certain position in an existing taxonomy hierarchy. In [9], a keyword-based vector space model is used to calculate the similarity of Web pages to the seed URLs, and if the similarity is above a certain threshold, the pages are downloaded and indexed, and their out-going links are followed.

A focused crawler [10] based on context graphs is proposed by so that the crawler can extract information about the context within which desired documents are usually found. A set of classifiers are then trained to classify in-degree Web pages according to an estimation of their relevance to the topic. The relevance estimation then navigates the crawler towards desired documents.

Crawlers with a probability model are used for calculating priorities, which combines Web page content-based learning, URL token-based learning, and link-based learning [11]. In a later work, [12] takes into account the users' access behavior and re-tunes the previous model to connect this behavior with the predicate satisfaction probability of the candidate Web pages waiting to be crawled.

An interesting "reversed" approach is proposed in [13], which suggests a given scientific document from a digital library be used as an input to the focused crawler. The main title and reference titles of the document are extracted and used to train a classifier to learn topical knowledge. The crawler is then guided by such knowledge to discover other topic-relevant documents on the Web.

More up-to-date reviews of focused crawling algorithms are presented in [14] and [15]. In [14], five different methods are implemented and evaluated within a unified evaluation framework on small and large datasets.

Here we discuss two studies that bear similarities to ours. The HPSearch and Mops presented in [16] support the search for research papers *close to* the homepages of certain scientists. However, their system does not investigate the issues of document harvesting for digital libraries for different publishing venues. Furthermore, our system outperforms theirs in terms of the percentage of correct homepages returned. In a more recent study [17], a Paper Search Engine (PaSE) is proposed, which uses citation information to locate online copies of scientific documents. While their study addresses a different research question, the PaSE system employs similar heuristics as we do to favor certain out-going links in order to quickly locate academic papers.

3. SYSTEM DESIGN

3.1 System Overview

We develop an automated system in which document metadata is used to automatically locate the homepages of the authors and focused crawl these homepages with the intent of finding missing documents. Our system, shown in Figure 1, consists of a Homepage Aggregator and a smart Focused Crawler.

The system accepts a user's request to harvest the desired papers published in a specific venue (e.g. a conference or a journal). The Homepage Aggregator will query a Public Metadata Repository and extract useful metadata heuristics to assist in quickly and accurately locating URLs of the authors' homepages. A list of such URLs will be inserted into the Homepage URL Database. The Crawler uses focused crawling techniques to search the domains for desired publications. It accepts the seed URLs as an input and uses them as starting points for the crawl. The Crawler uses anchor text to determine link priorities and quickly navigates through the websites using to get to the desired academic papers. The harvested documents will be stored in the Document Database.

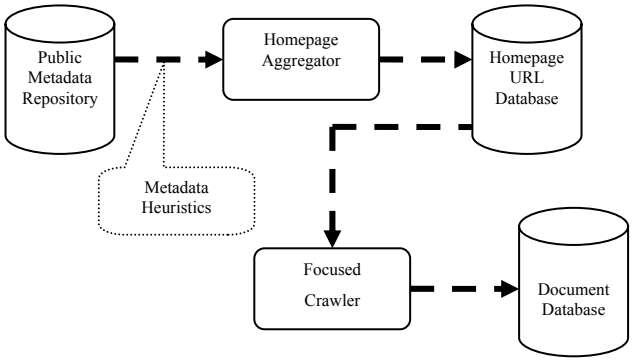

Figure 1. System Architecture

3.2 Using Metadata to Locate Homepages

Crawling authors' homepages first requires the system to be able to locate such websites quickly and accurately. A study of the literature indicates that personal website and homepage finding have been studied a lot since the birth of WWW. In [18], the authors present AHOY! as the first working system for personal homepage finding, which can filter irrelevant pages based on pattern matching heuristics. Later, the TREC (Text REtrieval Conference) hosted the task of Web homepage finding in 2001 and its subsequent years, and algorithms based on link analysis, linguistic cues, and machine learning etc. are proposed [19, 20, 21]. Examples of current working systems include HomePageSearch (hpsearch.uni-trier.de) which is a Homepage Aggregator mainly for computer scientists, and compiled directories (e.g. Google Directory)

See Figure 2 for the architecture of the Homepage Aggregator component.

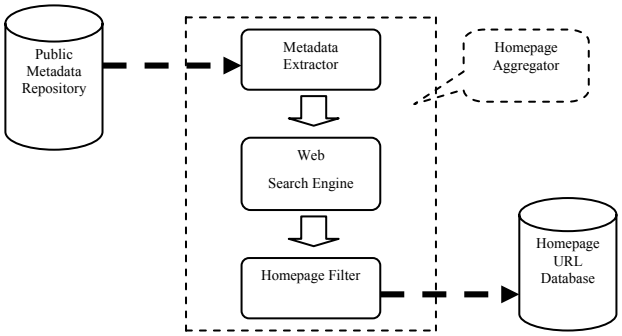

Figure 2. Architecture of the Homepage Aggregator

The goal of the Homepage Aggregator is to look for homepages of the authors and save them as seed URLs to feed the Focused Crawler. First it queries the Metadata Repository and retrieves the document metadata. For each author, it extracts from metadata a value pair of (*N*, *P*), where *N* is the name of the author and *P* is the name of the venue (with a number of variations) in which the paper is published. A list of such pairs is then submitted to a Web search engine. Pages returned by the search engine will go through a Homepage Filter where we use metadata heuristics to remove false positives (pages that are not likely to be the homepages of the authors) and disambiguate among namesakes, if there is any. Different priority weights are assigned to the remaining pages according to their likelihood of being the homepage of the author. The more likely it's the homepage of the author, the higher priority it receives. Eventually the page with the highest priority weights will be inserted into the Homepage URL Database, and will be crawled later.

Recall that we extract from metadata a pair value of (*N*, *P*). Now let *U* be the URL and *T* be the title of a Web page *P* returned by the Web search engine. When there are more than two authors for the same paper, assume *Ui* are the URLs of the homepages of other authors already found by the system. We have incorporated the findings in [16] about major characteristics of personal homepages. The metadata heuristics employed in the Homepage Filter are explained in Table 1.

Table 1. Heuristics Employed in Homepage Filter

Function	Heuristic Rules
Remove false positives	● Remove *U* if *U* or *T* indicates a publisher's website. ● Remove *U* if *U* or *T* indicates a digital library. ● Remove *U* if *U* points to a file other than .htm/.html
Disambiguate between namesakes	● Choose *U* among the candidates if *U* is in the same domain as *Ui*. ● Remove *U* if its parent-domain is already found by the system.
Assign priority	● *U* receives high priority if *T* contains *N* and any of the following: *homepage (home page), web (website), research, publication, papers*. ● *U* receives medium priority if *T* contains any of the following: *homepage (home page), web (website), research, publication, papers*. ● *U* receives low priority when neither one of the above two rules is fired.

3.3 Crawler Architecture

The Focused Crawler crawls web pages, using heuristics to quickly navigate to the publications. The architecture of the component is shown in Figure 3.

The crawler accepts two primary sets of inputs that vary for each crawl. The first is a set of seed URLs that are the starting points of the crawl. These are added to the crawl queue at low priority. The second set of inputs is a collection of domain names that the crawler is permitted to crawl.

Once the seed URLs are entered into the queue, the crawler threads are started. Each thread gets one URL from the priority queue, and downloads the page that it points to.

After a page is downloaded, the out-going links are examined and those matched with the ignored list are removed, either because they are out of the target domain or because their MIME types are not processed by the crawler. At this point, if a PDF/PostScript document is found, it will be inserted into the Document Database. The rest of the out-going links will each be classified as high, medium, or low priority, and inserted into different priority queues.

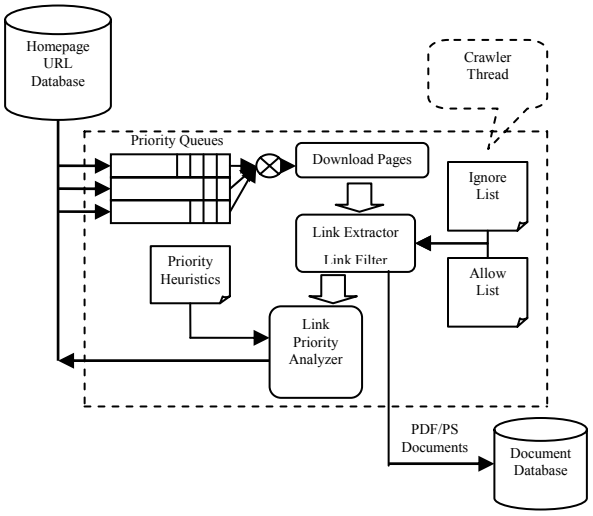

Figure 3. Architecture of the Focused Crawler

In order to concentrate or limit the crawls towards only desirable content, the crawler is provided with three lists for reference. The contents of the lists may be changed depending on the types of domains being crawled.

The Ignore List is a set of file types that are to be ignored by the crawler. The most common types of URLs that are ignored by the crawler are links to image files. The list can also include parts of the domain(s) being crawled, which the crawler is not supposed to visit. Table 2 shows a sample Ignore List.

Table 2. Sample Ignore List

File Types	.jpg, .bmp, .gif, .png, .jpeg, .mpg, .mpeg, .avi
Domains	http://clgiles.ist.psu.edu/picture.html
	http://clgiles.ist.psu.edu/courses.html

Files of type JPG, BMP etc will be ignored during the crawl. Also any outgoing links to pages within the ignored domains will not be considered for crawling.

The Allow List on the other hand is a collection of domain names that make up the crawl space of the crawler. Links pointing outside the specified domains are ignored by the crawler (unless they are determined to be research documents). This list is useful to limit the breadth of the crawl to only those domains that are of interest. Table 3 shows a sample Allow List.

Table 3. Sample Allow List

Domains	http://clgiles.ist.psu.edu

So the link http://clgiles.ist.psu.edu will be considered for crawling if it's discovered.

Priority lists contain a set of keywords and their assigned weights that are used to determine the priorities of the extracted links. The links will be visited by the crawler in the order of their assigned priority.

The Crawl Queue holds the discovered URLs that are yet to be crawled. This queue consists of three sub-queues: High-priority, Medium-priority and Low-priority queue. The Low-priority queue is the default queue. The seed URLs are entered into this queue.

We adopt a simple yet very effective heuristics to make the priority classification based upon the likelihood of the link eventually leading to academic publications.

We first train a classifier with data collected from two publishing venues: the Very Large Data Bases (VLDB) Conference and the Text REtrieval Conference (TREC). Several crawls are carried out with a breadth-first policy. The logs of the crawls are analyzed and a traverse tree is generated for each of the crawl that indicates the URLs visited and the link path that is followed by the crawler to reach the desired publications.

Consider a small website having 11 pages as shown in Figure 4.

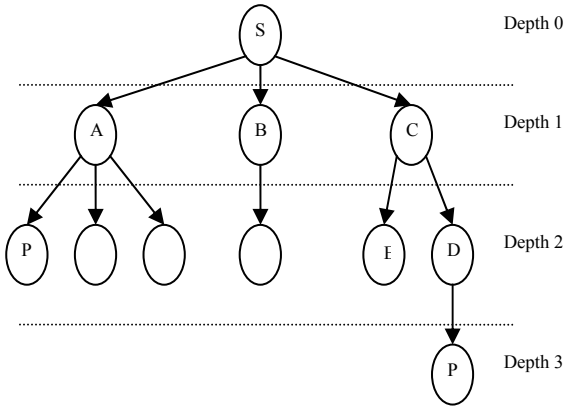

Figure 4. Sample Website

The circles represent URL's in the website and the arrows are the hyperlinks from one page to another. The link structure shown is that which is followed by the breadth-first crawler to visit each URL. All other links such as those that may point outside the domain are ignored in the above diagram.

The node marked with 'S' is the seed or start URL. The nodes marked with 'P' are research document files that are detected by the crawler. Now the links that are of interest to us are S→A→P and S→C→D→P. The anchor text contained in these links 'SA', 'AP', 'SC', 'SD', 'DP' is extracted and marked as 'interesting'. The text in the remainder of the links is also noted, but goes in 'not interesting' set.

Similar analysis is done on all the logs that are generated by the breadth-first crawl. All the keywords that are commonly occurring in the "interesting" class and not so commonly occurring in the "non-interesting" class are extracted. Weights are assigned to each of these keywords depending on their placement in the link structure. The keywords closer to the documents are given more weight that those closer to the seed URL. For e.g. keyword 'SA' has a lesser weight than keyword 'DP' as 'DP' is closer to P than to S as opposed to 'SA'.

The formula for calculating keyword weight is:

$$W(OT_{o \to q}) = D(Q) / D(P) \quad \text{(I)}$$

where $OT_{o \to q}$ is the anchor text of the out-going link from page O to page Q; P is the desired academic paper found by following the link from O to Q; $D(P)$ denotes the distance (number of hops) between P and the starting URL S on the path

$S \rightarrow \ldots \rightarrow O \rightarrow Q \rightarrow \ldots \rightarrow P$; $D(Q)$ denotes the distance between Q and the starting URL S on the path $S \rightarrow \ldots \rightarrow O \rightarrow Q$.

Now that a list of anchor texts and their corresponding priority weights has been compiled during the training process, we can classify each of them into different priority categories according to the weights. Table 4 shows a few samples extracted from our list.

Table 4. Sample Anchor Texts

Priority	Anchor Texts
p_High	volume, pub, paper, conf, journal, content, program, research, list
p_Medium	topic, faculty, people, group, lab

We now need to consider how to prioritize out-going links that are more likely to lead to desired academic publications. The anchor text in these links is compared against the weighted keywords. If any of the weighted keywords are present in the text, the comparison is considered to be successful. There are no keywords having more than one weight. The final priority of the link is calculated by the following function.

```
// Get_Priority(): Returns the priority for link L_T with anchor
text T which has weight W_T.
// Low=0, Medium=1, High=2 (for weight and priority)
Get_Priority {
If W_T = 0 and (Priority(Parent(L_T)) > 0 then
        Priority(L_T) = Priority(Parent(L_T)) -1;
Else if W_T > 0
        Priority(L_T) = W_T ;
End IF
Return (Priority(L_T));
}
```

The priority of a link may also depend on the priority of its parent. This is mainly due to the fact that not all the links that emerge from a page with a medium or high priority may lead to a research document. For e.g. in Figure 4 the node 'C' will be crawled with a medium priority, however only node 'D' leads to a research document. The priority of the node 'E' is thus reduced to low as it will not have a weighted keyword attached to it and that of 'D' is increased to high. The priorities of links thus established are used to insert the link in the proper priority queue for crawling. In order to achieve high efficiency, the crawler spawns multiple threads which will be fed with URLs on the descending order of priority. When there is no URL left in the priority queues and no crawler thread is currently running, the crawling task is finished.

4. RESULTS AND DISCUSSION

We have collected data from two Computer Science publication venues: the ACM SIGMOD International Workshop on the Web and Databases (WebDB), first held in 1998 and then each year in conjunction with the annual ACM SIGMOD Conference, and Journal of Artificial Intelligence Research (JAIR), which was established in 1993 both as an electronic scientific journals and a hard-copy semiyearly published by AAAI Press. We choose these two venues because both of them are highly selective venues with less than a 25% acceptance rate and we want to observe if there is a major difference of performance between conferences and journals.

We have extracted the metadata of WebDB and JAIR from the DBLP repository. By analyzing these metadata, we successfully identify the 593 unique authors who have in total published 289 papers in either one of these two venues during the period from 1998 to 2004. Please see Table 5 for more details of the dataset.

Table 5. Statistics of the collected data

Year	WebDB		JAIR	
	Unique Authors	Publication	Unique Authors	Publication
1998	32	13	40	20
1999	51	17	50	28
2000	61	20	33	20
2001	51	18	45	25
2002	47	17	64	27
2003	56	17	72	30
2004	51	16	57	21
Total	285	118	308	171

In order to examine whether our approach is effective in recovering those missing documents from a digital library, we use the CiteSeer Scientific Digital Library as another data source. Cross-referencing the metadata of each of the two venues from DBLP, we successfully identified 30 out of 118 (25.42%) WebDB papers and 46 out of 171 (26.90%) JAIR papers that are not indexed by CiteSeer (see Figure 5 for details). This is done by exact title-matching between the records in the DBLP metadata repository and the CiteSeer document archive.

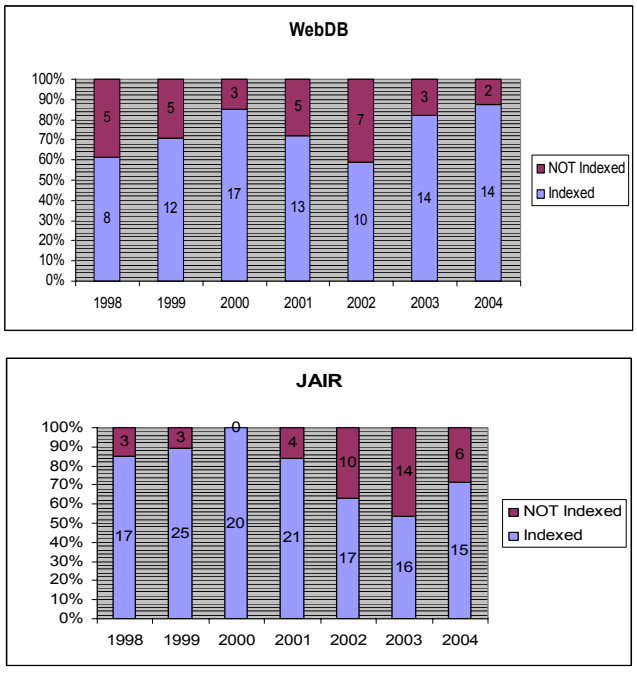

Figure 5. Coverage of the two venues by CiteSeer

The metadata extracted from DBLP are also used as heuristics to locate the homepages of the 593 authors. The name of the author

and the corresponding venue (with a number of variations) are submitted to Google API and the first 10 URLs returned are parsed automatically by the Homepage Filter component. Using the heuristics discussed in the previous section, we assign priority weights to each of the URLs. For each author, URLs with the highest priority weights are inserted into the URL Database and crawled by the Focused Crawler at a later stage.

We have manually examined the records in the URL Database in order to evaluate the effectiveness of the Homepage Aggregator. In total, homepages of 539 authors (90.89%) have been found. Details about the 54 authors whose homepages cannot be found by the system are shown in Table 6. Here we define Non-U.S. authors to be those whose affiliations are not currently in the States.

Table 6. Number of authors whose homepages are not found

	WebDB	JAIR
U.S. Authors	13	6
Non-U.S. Authors	25	10
Total (Percentage)	38 (13.33%)	16 (5.19%)

There are only 2 papers ([22], [23]) of which all the authors' homepages are not found by the system, which account for less than 1% of the 289 papers in our data set. In other words, although the system fails to locate the homepages of about 9% of the authors, it is not a major performance impact on the document recall and the crawler should still be able to find 99.31% of all the papers.

For the cases where the system fails to locate some of the homepages, we notice that most of the 19 U.S. authors whose homepages are not found were actually in their graduate programs when they co-authored the paper, and their Web presences seem to have disappeared after graduation. In addition, there's a significant difference between the numbers of U.S. and non-U.S. authors whose homepages cannot be found, with non-U.S. almost twice the number of U.S. authors. Since this is our initial attempt limited to only the domain of computer science, whether this difference holds true for other disciplines and the reason behind remain an open question. Finally, there are several cases where the homepages of those with famous names actually show up instead of the desired authors. For example, a search via Google API for the first author in [24] returns the homepage of a comic artist. The top 5 websites for *George Russell*, the first author of [25], happen to belong to that of a famous Jazz musician. There are also a few cases where the search engine actually returns the homepage of the co-author instead of the author himself, because the author's name is listed on the co-author's page as a collaborator and the co-author's page receives a higher page ranking. All these indicate that the disambiguation capability needs to be improved.

4.1 Finding Desired Academic Publications

When the crawl is finished, we manually examine the downloaded PDF/PostScript documents in order to evaluate the performance of the crawler. In total, the crawler has acquired 236 out of the 289 papers (81.66%) published in WebDB (100 out of 118, 84.75%) and JAIR (136 out of 171, 79.53%) from 1998 to 2004. For details of the results for each venue, please see Figure 6 and 7.

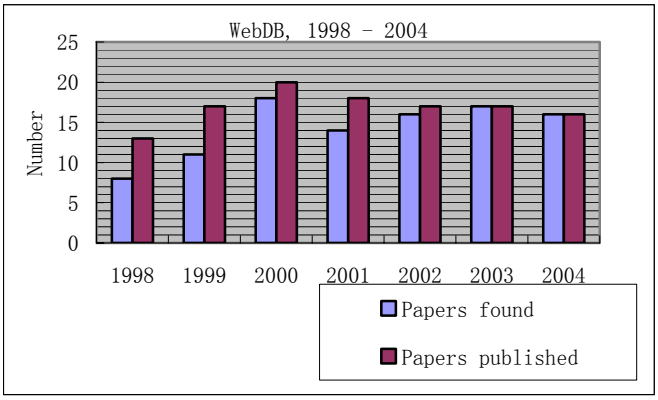

Figure 6. Number of WebDB Papers

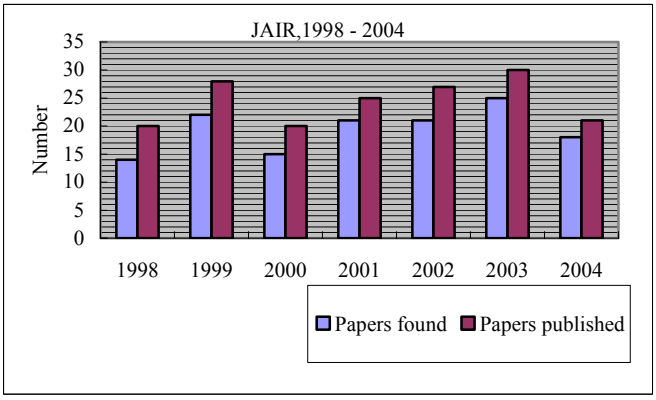

Figure 7. Number of JAIR Papers

Here we adopt one of the performance metrics, *recall level*, first proposed in [16] and used in [17]. *Recall level* is defined as:

$$\rho(i) = | S(i) \cap T | / | T |$$

where *S(i)* is the set of documents downloaded by the crawler during a crawl on the dataset of a calendar year *i; T* is the set of desired documents, which in this study are the papers published by a specific venue in the same calendar year. This measure represents the capability of the system to capture desired academic papers.

Overall, our system has achieved a *recall level* of 0.8475 for WebDB and 0.7953 for JAIR documents. See Figure 8 for more details.

It's interesting to note that while the *recall level* of WebDB is constantly increasing until reaching 1.0 in the last two years, the *recall level* of JAIR seems to fluctuate around 0.8 over the 7-years period. We find that 29 out of the 35 (82.86%) JAIR papers not found by the system are actually downloadable via a link from the authors' homepages to the publisher's website. Yet we miss these papers simply because we limit our crawler not to go beyond the domain of authors' homepages. We believe that a more sophisticated domain restriction for the crawler can be easily employed in order to achieve an even higher recall level.

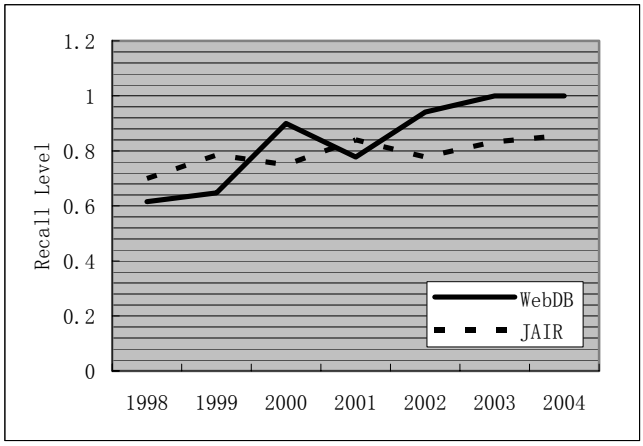

Figure 8. Overall Recall Level, 1998 - 2004

We calculate the *recall level* for the documents published in WebDB and JAIR yet missing from CiteSeer's collection (see Figure 9). In this case, *S(i)* is the set of missing documents downloaded by the crawler, and *T* is the set of the papers not indexed by CiteSeer and missing from the collection. On average, the recall level has achieved 0.78 for WebDB and 0.72 for JAIR. Especially WebDB's recall level is constantly increasing, reaching 1.0 for the last three years. This proves that it's practical to harvest the missing documents for a given publishing venue.

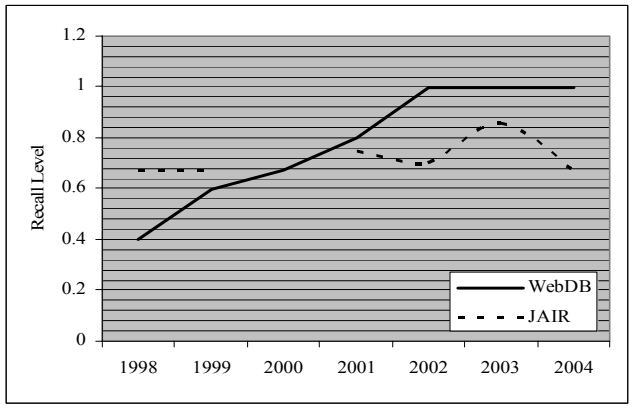

Figure 9. Recall Level for the Missing Documents

The trends shown in Figure 8 and 9 seem to indicate that a rising number of academic papers have been put online, especially in and after the year 2000. However, it's interesting to note that it seems conference/workshop authors favor putting their publications on their homepages, while journal authors don't. Due to the limited size of our sample, we feel this is an open question to be answered with more data across multiple venues.

4.2 Crawler Comparison: BF Crawler

In order to further evaluate the performance of our system, we also compare our work to other crawling approaches. First we crawled three conference websites using our system and a breadth-first (BF) crawler. Figures 10, 11 and 12 show the results of crawls on different conference websites. The BF crawls are shown by the dashed line while the results of the focused crawler are shown by the solid line on the figures. The horizontal axis indicates the number of pages crawled and the vertical axis represents the number of research documents found by searching those pages. The number of documents found is a cumulative sum of all PDF, PS and GZ files found on those sites. Since they may contain duplicate files or the same content in different file types, the numbers shown do not indicate unique papers. The number of pages crawled does not include academic papers. The same crawl restrictions applied to both the crawlers.

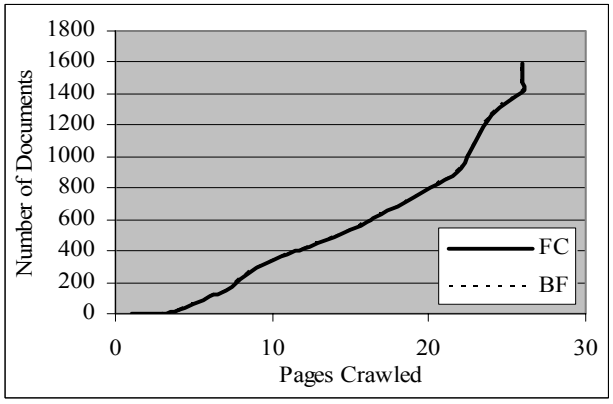

Figure 10. ACL Conference Crawl

Figure 10 shows the crawls done on parts of the Association for Computational Linguistics (ACL) conference website. The total number of pages crawled on this site were less than 30. Both crawls overlap which indicates that there is virtually no difference between the document detection rate of the BF crawler and our focused crawler. For such a small website, both crawlers detect the same number of documents after crawling the same number of pages on the website.

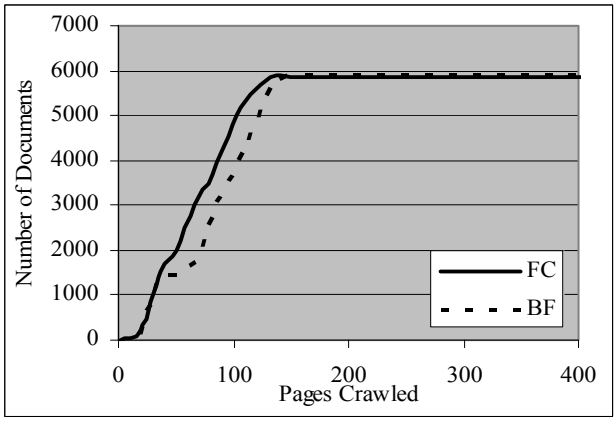

Figure 11. TREC Conference Crawl

Figure 11 shows the crawls done on the Text Retrieval Conference (TREC) pages. Here the total of pages crawled is about 1000 (only first half of the crawl is shown in the graph). Both crawlers start detecting documents at the same rate. After detecting around 1393 documents (35 pages crawled) the document detection rate of the focused crawler becomes slightly better than the BF crawler. Although the difference is not very significant, the focused crawler does detect the research documents slightly earlier in the crawl as compared to the BF crawler. The BF crawler detects the same amount of documents (4800 documents) as the focused crawler but after crawling 20-30

pages more than the focused crawler. The total number of documents found by both the crawlers is around 6000.

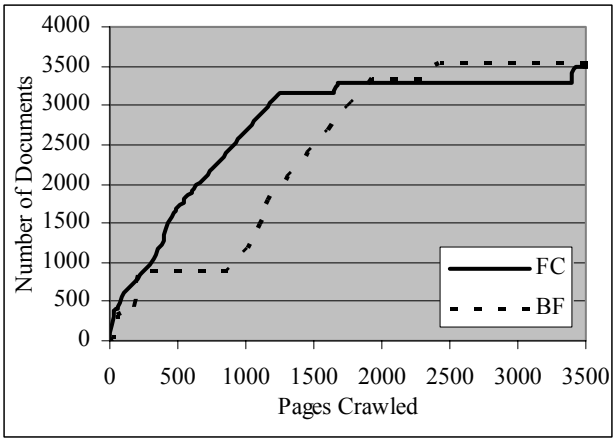

Figure 12. VLDB Conference Crawl

The crawls performed on the Very Large Database (VLDB) conference pages as shown in Figure 12 indicate that the focused crawler detects the documents much earlier in the crawl. Here the total number of pages crawled is about 3500. Approximately 28% (1000 out of 3500) of the documents are located by both the crawlers after crawling around 8.5% (300 out of 3500) of the domain. At this point the focused crawler continues to locate more documents while the BF crawler does not uncover any new documents until 28% (1000 out of 3500) of the total crawl. 85% (3000 out of 3500) of the documents are located by the focused crawler after completing just 33% (1189 out of 3500) of the total crawl, while the breadth first crawler locates the same amount of documents after completing 50% (1781 out of 3500) of the total crawl. Towards the end of the crawl the breadth-first crawler detects more papers as compared to the focused crawler. It takes the focused crawler around 1000 more pages of crawls until it makes up the difference. This seems to be due to the lack of keywords associated with the links that eventually led to the documents. The focused crawler evaluates other papers that have a higher priority values before eventually discovering the remaining documents.

The behavior of the BF crawler is consistent for all the three crawls. Most of the documents located were in crawl depths 2, 3, 4 and 5. The BF crawler detects them after completing search of the previous crawl depths. As the focused crawler prioritizes the links for crawling, the higher depths with more priority are crawled before the lower depths with less priority.

The above experiment indicates that the document harvest rate is almost the same for smaller websites. The difference becomes apparent when the size of the website being crawled is large. The focused crawler is able to detect the documents much earlier in the crawl as compared to the BF crawler. Since the crawls are not terminated early for the focused crawler, the number of documents found and the relevance of documents are same for both the crawlers. Therefore as the size of websites being crawled increases, the focused crawler detects more documents earlier during the crawls as compared to the BF crawler.

We assess the crawler's capability of harvesting academic publications in a more general sense which is not only limited to a specific venue. We have manually examined the first 500 PDF/PostScript documents found by the two crawlers, classified the documents into academic publications which are desirable (papers published in conferences and journals; technical reports; degree thesis, etc.), and non-publication documents which are considered *noise* for a publication collection (course material; presentation slides; project schedule; etc.) Percentage of both categories is compared side-by-side and shown in Figure 13. Our crawler has outperformed the breadth-first counterpart by having much less of this *noise*.

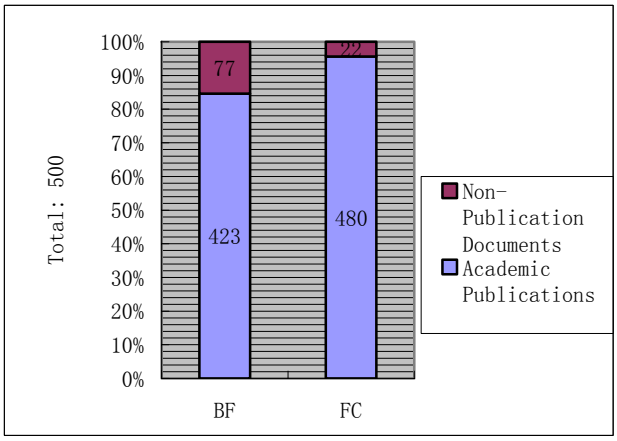

Figure 13. Composition of the First 500 PDF/PS Documents

4.3 Crawler Comparison: Nutch Crawler

We compare the performance of our system with Nutch (http://www.nutch.org/docs/en/), an open source Web crawler and search engine built upon Lucene. In our experiment, we run the Nutch crawler on the official websites of WebDB and JAIR, and identify those papers published between 1998 and 2004 from the downloaded documents. We then compare the number of papers harvested by Nutch and FC crawler (see Figure 14 for details). Results show that guided by certain heuristics, crawling authors' homepages can actually achieve almost the same recall level as crawling publishers' websites.

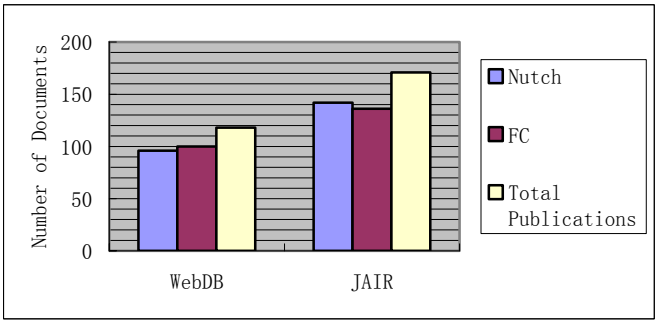

Figure 14. Comparison between Nutch and Focused Crawler

Figure 15 indicates the progress of the crawls conducted by both the Focused Crawler and the Nutch Crawler on the ACL conference website. The documents found are of PDF and PS only. The focused crawler starts discovering documents earlier in the crawl and the process continues gradually. Nutch on the other hand discovers most of the documents after crawling around 84% (22 out of 26) of the website.

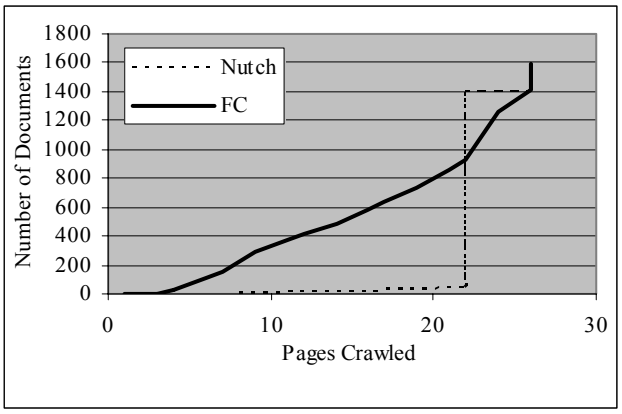

Figure 15. Crawling ACL Conference Websites

Documents found during the ACL conference crawl are classified into two categories: relevant (i.e. academic publications) and non-relevant (non-publication). Figure 16 shows the number of documents in each category. Note that determining documents' relevancy is an offline process. Here *R* indicates relevant and *NR* indicated non-relevant documents.

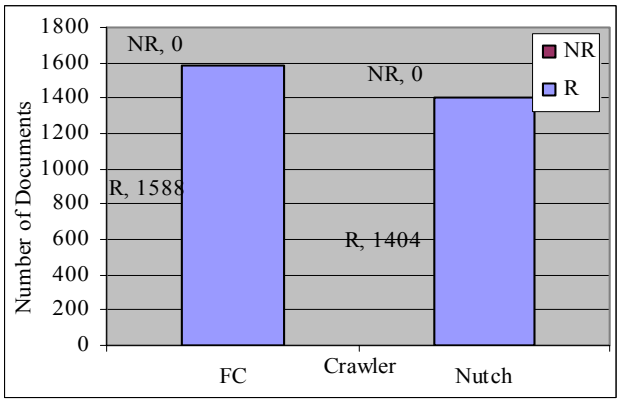

Figure 16. Relevancy of the ACL Conference Crawl

Figure 16 indicates that all the documents (PDF and PS) found by both the crawlers are academic publications (thus *NR* = 0). However, the 184 documents Nutch failed to detect are determined to be all relevant research publications.

The same comparison is also conducted by crawling the official WebDB conference websites. Figure 17 shows that the Focused Crawler starts detecting desired documents at an earlier stage as compared to the Nutch crawler. Yet due to the small number of pages crawled, a rigorous comparison cannot be made in this case.

Figure 18 shows that the focused crawler locates two more academic publications than the Nutch crawler, both of which are marked as relevant documents.

5. CONCLUSION AND FUTURE WORK

We have shown the feasibility of using authors' homepages as alternative online resources to harvest the academic papers missing from a collection of digital libraries, as well as the techniques to maximize the crawler's performance in doing so. We have designed and implemented a heuristic-based system which utilizes document metadata to accurately locate authors' homepages and performs a focused crawling to quickly navigate to the desired publications. Evaluation has been conducted using a large dataset collected from several publishing venues in the Computer Science domain, and detailed results are presented and discussed.

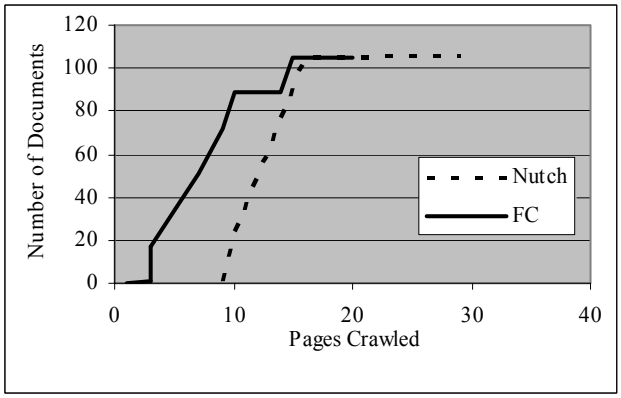

Figure 17. Crawling WebDB Conference Websites

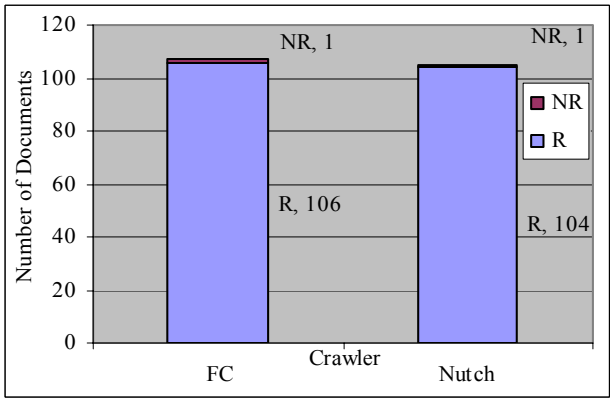

Figure 18. Relevancy of the WebDB Conference Crawl

For the academic venues investigated in this study, we are able to fill many of the missing documents in the CiteSeer digital library.

The designed focused crawling technique efficiently locates desired publications on authors' homepages as well as conference websites. The Homepage Aggregator detects homepages well and the Focused Crawler outperforms the baseline crawler in a number of measures.

Future work includes a more rigorous disambiguation scheme for the Homepage Aggregator and a more sophisticated weighting scheme for the Focused Crawler. In addition, we are now developing a training process for the crawler to learn the URL patterns of alternative resources other than author homepages, such as institutional archives. Also, the automation of the process cycle of crawling, log analysis, and heuristics generation can help search engine based digital libraries scale and significantly reduce costs. The actual URL of the web pages can also be used to assist in priority assignment instead of just using the anchor text of the link. A comparison of this approach to techniques other than a Breadth-first crawl is currently underway. Furthermore, we plan to evaluate the validity of this approach by expanding our experiment on to disciplines other than the Computer Science

domain. We believe our study and its consequents will shed lights on the question of finding missing papers for our digital library, or "what's there and what's not".

6. ACKNOWLEDGEMENTS

We gratefully acknowledge P. Mitra and the anonymous reviewers for their comments, I. Councill and P. Teregowda for their work on the CiteSeer metadata, and E. Maldonado and D. Hellar for the crawl list. This work is partially supported by Microsoft.

7. REFERENCES

[1] De Bra, P., Houben, G., Kornatzky, Y., and Post, R Information Retrieval in Distributed Hypertexts. In *Proceedings of the 4th RIAO (Computer-Assisted Information Retrieval) Conference,* pp. 481-491, 1994.

[2] Cho J., Garcia-Molina, H., and Page, L. Efficient Crawling Through URL Ordering. In *Proceedings of the 7th World Wide Web Conference*, Brisbane, Australia, pp. 161-172. April 1998.

[3] Chakrabarti, S., Van den Berg, M., and Dom, B. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. In *Proceedings of the 8th International WWW Conference*, pp. 545-562, Toronto, Canada, May 1999.

[4] Giles, C. L. and Councill, I. G. Who gets acknowledged: Measuring scientific contributions through automatic acknowledgement indexing. In *Proceedings of the National Academy of Sciences* 101(51) pp. 17599-17604, Dec. 21, 2004.

[5] Najork, M. and Wiener, J. L. Breadth-First Search Crawling Yields High-Quality Pages. In *Proceedings of the 10th International World Wide Web Conference*, pp. 114-118, 2001.

[6] Page, L., Brin, S., Motwani, R., and Winograd, T. *The pagerank citation ranking: Bringing order to the web.* Technical report, Stanford University Database Group, 1998. Available at http://dbpubs.stanford.edu: 8090/pub/1999-66

[7] Menczer, F., Pant, G., Ruiz, M., and Srinivasan, P. Evaluating Topic-Driven Web Crawlers.' In *Proceedings of the 2001 Annual Conference of the Association of Computing Machinery, Special Interest Group in Information Retrieval*, 241-249. New Orleans, September 2001.

[8] Haveliwala, T. H. Topic-Sensitive PageRank. In *Proceedings of the 11th International World Wide Web Conference*, pp. 517-526. Honolulu, Hawaii, USA. May 2002.

[9] Mukherjea, S. WTMS: a system for collecting and analyzing topic-specific Web information. *Computer Networks* 33(1-6): 457-471, 2000.

[10] Diligenti, M., Coetzee, F.M., Lawrence, S., Giles, C. L., and Gori, M. Focused Crawling Using Context Graphs. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 527-534, 2000.

[11] Aggarwal, C. C., Al-Garawi, F., and Yu, P. S. Intelligent Crawling on the World Wide Web with Arbitary Predicates. In *Proceedings of the Tenth International Conference on World Wide Web*, pp. 96-105, 2001.

[12] Aggarwal, C. C. On Learning Strategies for Topic Specific Web Crawling. *Next Generation Data Mining Applications*, January 2004.

[13] Pant, G., Tsjoutsiouliklis, K., Johnson, J., and Giles, C. L. Panorama: Extending Digital Libraries with Topical Crawlers. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries*, pp. 142-150, 2004.

[14] Menczer, F., Pant, G., and Srinivasan, P. Topical Web Crawlers: Evaluating Adaptive Algorithms. *ACM TOIT* 4(4): 378-419, 2004.

[15] Pant, G., Srinivasan, P., and Menczer, F. Crawling the Web. In M. Levene and A. Poulovassilis, eds.: *Web Dynamics*, Springer, 2004.

[16] Hoff, G. and Mundhenk, M. Finding scientific papers with homepagesearch and MOPS. In *Proceedings of the Nineteenth Annual International Conference of Computer Documentation, Communicating in the New Millennium*, pp. 201-207. October 21-24, 2001, Santa Fe, New Mexico, USA.

[17] On, B. and Lee, D. PaSE: Locating Online Copy of Scientific Documents Effectively. In *Proceedings of the 7th International Conference of Asian Digital Libraries (ICADL)*, pp. 408-418. Shanghai, China, December 2004.

[18] Shakes, J., Langheinrich, M., and Etzioni, O. Dynamic Reference Sifting: a Case Study in the Homepage Domain. In *Proceedings of the Sixth International World Wide Web Conference*, pp. 189-200, 1997.

[19] Xi, W. and Fox, E. A. Machine Learning Approach for Homepage Finding Task. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pp. 686-698, 2001.

[20] Anh, V. N. and Moffat, A. Homepage Finding and Topic Distillation using a Common Retrieval Strategy. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, 2002.

[21] Ogilvie, P. and Callan, J. Combining Structural Information and the Use of Priors in Mixed Named-Page and Homepage Finding. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003),* pp. 177-184, 2003.

[22] Sundaresan, N., Yi, J., and Huang, A. W. Using Metadata to Enhance a Web Information Gathering System. In *Proceedings of the Third International Workshop on the Web and Databases (WebDB 2000)*, pp. 11-16, 2000.

[23] Flesca, S., Furfaro, F., and Greco, S. Weighted Path Queries on Web Data. In *Proceedings of the Fourth International Workshop on the Web and Databases (WebDB 2001)*, pp. 7-12, 2001.

[24] Ruiz, A., López-de-Teruel, P. E., and Garrido, M. C. Probabilistic Inference from Arbitrary Uncertainty using Mixtures of Factorized Generalized Gaussians. *Journal of Artificial Intelligence Research (JAIR)*, Volume 9, pp. 167-217, 1998.

[25] Russell, G., Neumüller, M., and Connor, R. C. H. TypEx: A Type Based Approach to XML Stream Querying. In *Proceedings of the Sixth International Workshop on the Web and Databases (WebDB 2003)*, pp. 55-60, 2003.