Evaluating Crawling Efficiency Using Different Weighting Schemes with Regional Crawler

Pirooz Chubak
Department of Computer Engineering
Sharif University of technology
Tehran, Iran
chubak@ce.sharif.edu

Milad shokouhi
Department of Computer Engineering
Bu-Ali Sina University
Hamedan, Iran
shokouhi@ce.basu.ac.ir

Abstract – **Regional Crawler has been introduced as an intelligent crawler with a new strategy for crawling WWW. The basic idea behind this crawler is to gather users' interests in a specific region and look for them in pages on WWW. Regional Crawler is a Topic-Specific crawler using Best-First algorithm for crawling. It finds and updates pages that are more relevant to a specific topic sooner and more often. This causes a significant reduction in search engine indexes and saves noticeable amount of time and memory.**
In this paper, we have implemented three different weighting schemes- "Lnu.ltu","tf.idf vector model" and "nxx.nxx"- and have tested our Regional Crawler using each weighting scheme and compared their results.

Keywords – **Regional Crawler, Intelligent Crawler, Web Information Retrieval, Weighting Models.**

I. INTRODUCTION

A. The Internet and Search Engines

Being used by most of the internet users each day, search engines are one of the most important services available on the web. Surveys show that 58.8% of internet users are looking for information [3], and most of them are using search engines for their purpose.

The number of web sites climbed from 3 millions in 1998 to more than 9 millions by the year 2002 [4] and as a result the number of web pages has increased from 2.1 billions in 1999 to 16.5 billions in 2003 [14]. On the other hand the number of Internet users, which was less than 160 millions in 1998, has grown to more than 600 millions by the year 2002 [9] and this trend is increasing day by day.

These statistics clarify the importance of a good search engine for answering the users' desires and queries in spite of the great amount of information available on the web.

Search engines take users' queries as input and produce a sequence of URLs that match the queries according to a rank they calculate for each document they have indexed [6 and 7].

B. What are crawlers?

Web crawling is a process to collect the web pages that are interesting to search the engine and usually a challenging task [13]. Web crawler is a program that traverses the internet automatically by retrieving a web page and then recursively retrieving all linked pages [2 and 7].

It takes weeks and months to crawl the entire web because of huge amount of information on it [1 and 3]. Even the largest search engines, like Google and Altavista, cover only limited parts of the web and much of their data are out of date for several months of the year [10]. In February 2004, Google announced that it has covered "6 billion items": 4.28 billion web pages, 880 million images and 845 million Usenet messages [14]. Therefore, it will not be possible to cover and update the rapidly changing information such as news that changes hourly or daily. Most of the recent works done on crawling strategies attempt to minimize the number of pages that need to be downloaded, or maximize the benefit obtained per downloaded page [10].

C. Regional Crawler Strategy

As the crawling process is very critical and time consuming, crawlers must be optimized to get the most beneficial pages from the Web. Beneficial pages are the ones that satisfy the needs of a larger set of users. Regional Crawling [29 and 30] strategy is a Best-First strategy that was designed to achieve this goal.

In opposed to other common crawling strategies like BFS and DFS that usually work in a predefined sequential manner, crawling path for Regional Crawlers cannot be anticipated, because they choose the next page for crawling intelligently.

For Regional Crawler, the crawling path (the sequence of URLs visited by the crawler) is based on common interests of users in certain regions. The needs and interests are determined according to common characteristics of the users such as geographical location, age, membership and job. The more a document shares the common interests of a region, the more chance it has for being crawled in the next steps.

II. RELATED WORK

Searching and looking for special kinds of information has always been a critical task, since the inception of the WWW. So far, many different techniques for information extraction have been designed and some of them are widely used in practice. By the rapid growth of web and the unprecedented challenges for general purpose crawlers, the need for Topic-Specific crawlers became obvious. *Focused Crawler* [28] was introduced by Chakrabarti in 1999. The goal of a focused crawler is to selectively seek out pages that are relevant to a predefined set of topics. Instead of extracting so many documents from the web without any priority, focused crawler would find the links that are likely to be most relevant for crawl leading to significant save in hardware and network resources and helping to keep the crawl more up to date. F. Menczer et al evaluated a Topic-Driven web crawler [12] and compared their crawler with different crawling strategies in their paper. Jason Rennie and Andrew McCallum used Reinforcement Learning [8] in crawling [23]. They measured the probability that a hyperlink in a page is likely to be linked to a relevant page, according to the texts in hyperlink neighbourhood and as result their crawler crawls the link with the highest probability value. Their job is a part of CORA, a domain-specific search engine containing computer science research papers (McCallum et al. 1999). Another similar approach was a crawler Agent, implemented for finding biomedical information on the web. [26] There are some other similar experiments which measure the similarity of page contents with a specific subject using special metrics and reorder the downloaded URLs for the next crawl in [24] or even evaluate a learning scheme for identifying which URL the spider should crawl next in order to increase the efficiency in topic specific web resource discovery [25]. Bing Liu et al combined the crawling strategy with clustering concepts [27]. For each topic they first retrieve a specific number of top ranked retrieved pages from Google for that topic and then they extract some other keywords from them. For example pages returned by Google for Information Retrieval most have the word "indexer" inside. So, "indexer" is defined as sub-topic for topic Information Retrieval and the crawler would also look for pages which contains this word. As result, their crawler would return more number of pages and greater precision. Eventually, Chen et al developed a smart topic driven crawler which was using Genetic Algorithms in order to increase the crawling efficiency [11].

In contrast to these methods, Regional crawler uses a new crawling strategy. Instead of looking for infinite number of subjects and clustering them, it takes into account that some users might not be interested in all subjects but people in the same region usually share some common interests. People in US are likely to be interested in baseball but people in UK have greater interest in Soccer. People in CS department of universities are probable to look for computer related articles or software and hardware tools in opposed to staffs of a travel agency or an elementary school students. As conclusion by devoting a specific regional crawler for each region (each country for example), we provide some pages that are important for most of people in that region and on the other hand we would save space and Network resources.

III. REGIONAL CRAWLER ARCHITECTURE

In centralized search engines [5], there is a central URL store, which sends URLs to the crawler for processing and downloading. The mechanism that leads to the production of a list of weighted URLs to get downloaded, determines the crawling strategy.

Each region in regional crawler consists of one or more IP addresses that identify a group of users. A region's granularity could be as small as a LAN or as big as a country. Figure 1, depicts the main components of a Regional Crawler. In this figure, robots.txt Processor determines the set of permitted URLs under the given IP address and stores them in Valid URLs Queue. URL Ranker takes the set of valid URLs from the Valid URLs Queue and gives a rank to each unranked URL. URL rank is determined by applying a weighting scheme to the page related to that URL. Therefore, the URL rank is the weight of its corresponding page in Regional Crawler implementation. Page Processing Unit processes the pages that are pointed to by the URLs, extracts the keywords, and passes them to URL ranker for weighting. Weights reflect the closeness of a web page to the users' interests in a region. Therefore, a page may receive different weights from different Regional Crawlers, according to different region interests. Interests are retrieved from Interest Manager, which specifies the interests according to the known characteristics of each region manually, or by learning algorithms.

Manual User Interest Assignment is more suitable for situations where one or a handful of individuals are considered as a region. In this case, an administrator can assign the keywords representing the interests of the users to the IP address(es) of that region. Another scenario that might make sense to use this approach is when we are assigning the interest to a very large and diverse region like a country. In this case, the number of interests common to groups of users and the number of such groups are so big that considering any combination of them as regional representation would result in downloading a large number of pages. In such cases, an administrator can assign a few interests common to most users to this region in order to provide an efficient update of the pages that cover most common interests like news sites, financial facts, sports etc.

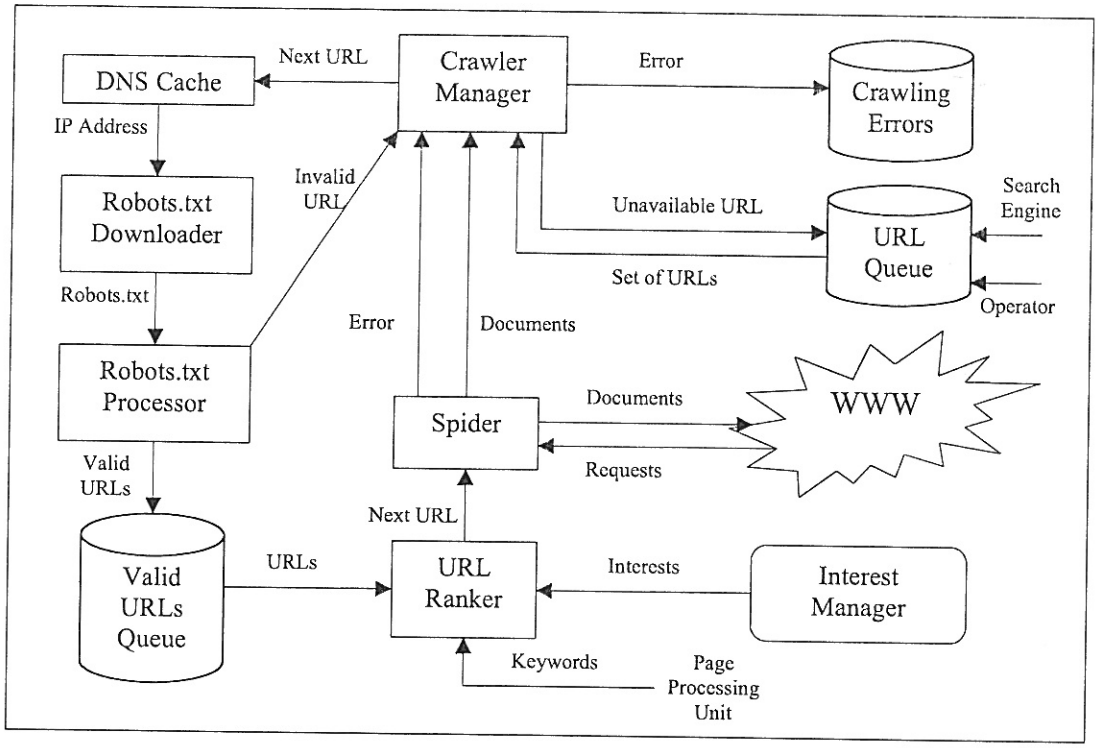

Fig. 1.Architecture of a Centralized Regional Crawler

For using learning algorithms in interest manager, we have to implement a learning method that learns the importance of keywords for describing the interests of a region. In its simplest form, it could be a standard Rocchio relevance feedback method [15].

URL Ranker provides the spider with an ordered list of URLs ordered by their priority for retrieval. In regional crawler, the order of downloading pages depends on users specific interests mapped to a region. For example, if in a region most users are much interested in soccer then regional crawler will crawl soccer related pages more often and if a university is considered as a region then scientific pages will be crawled more often for that region.

IV. IMPLEMENTATION AND RESULTS

In our past experiments, we implemented the Regional Crawler using nxx.nxx weighting scheme and tested it for different websites [30]. The precision of downloaded pages for Regional Crawler according to users' interests for that region was much more than a baseline crawler (BFS crawler in our test) extracted pages with the same weighting scheme. Here again we include some other results for comparison in the next subsection.

As we mentioned above, a Regional Crawler crawls on WWW according to the interests and needs of users of its region. These interests are usually terms that users look for in WebPages. For example users in region Iran are so probable looking for news about Iran, Tehran (the capital of Iran) and Middle East. We can manually assign these terms as user interests for region "Iran" so that Iran Regional Crawler knows these terms are important in pages and takes them as interest terms and uses them for assigning weight to WebPages. The highest weight page will then be crawled.

```
Crawl(Page)
    foreach link in Page.links
        linkPage ⟵ loadPage(link)
        linkPage.weight ⟵ calculateWeight(linkPage)
        if linkPage.weight > Page.weight
            Crawl(linkPage)
```

Fig. 2.General Algorithm for a Regional Crawler

Figure 2, shows the general recursive algorithm for Regional Crawlers. In this figure loadPage is a method that downloads a hyperlink and stores its body content and any links inside page body into linkPage. setWeight gives a weight to each linkPage according to its content. Best-First crawlers generally crawl the page with the highest weight in each step, therefore, using this strategy for crawling, our crawler always crawls the most similar page with user interests in the queue. So, if a link with higher weight than current page appears, it will have the highest weight, and the crawler starts crawling it instead. For calculating the page weight there are many weighting schemes in the literature each trying to maximize the precision of retrieval. "Precision" is the ration of number of relevant pages over number of retrieved pages, so a

higher precision indicates a better retrieval. We have implemented three different schemes for evaluating our crawler and also a baseline crawler with BFS crawling strategy. The schemes are nxx.nxx, tf.idf vector model and Lnu.ltu. In all of our experiments, we assumed the interest terms vector as {Iran, Tehran, Khatami (President of Iran), middleeast} with priorities 4, 2, 1, 1) for region Iran. In fact we have assumed that people are requesting Iran about four times they request middleeast. This assumption is not accurate but can give a good estimate.

Note that whatever the weighting scheme is, the BFS crawler collects the same page sequence and gives the same precision. But changing the weighting scheme can have a profound effect on crawling of Regional Crawler, since it decides its crawling sequence according to page weights.

In the following subsections we first introduce each weighting scheme and show the results for each, then we compare the weighting schemes.

A. nxx.nxx weighting scheme results

nxx.nxx is generally a weak weighting scheme based on [16]. It means the probability values of relevance and retrieval for a single page are not close. So a page which may be relevant to users interests may not be retrieved while other pages not relevant to the subject the user is looking for may get retrieved. We don't carry out any weight normalization in nxx.nxx method. "Normalization" is a method by which we reduce the effect of a parameter in weight calculation, so that we don't get abnormally high weights for some documents. This parameter is usually document length.

nxx.nxx simply assigns the sum of term frequencies in each page as its weight. Neglecting document frequency and length, the nxx.nxx scheme usually prefers long documents to short ones. We leave a comparison between these three weighting schemes in our tests for later subsections and briefly show the results of this scheme on Regional Crawler and BFS Crawler.

Table I, shows the average weight and precision for 200 WebPages downloaded by each crawler with the index page of gbgm-umc.org website which contains world news. The average weight in this table is not a suitable measure to decide about the performance of Regional Crawler. So we did the precision calculation by a number of users. Given the interest set for region Iran, users had to decide which pages are relevant to the interests and which ones are not.

TABLE I

nxx.nxx RESULTS

URL	http://www.gbgm-umc.org		
Number of Iteratio	213		
	Average Weight	Max Weight	Precision
Regional Crawler (nxx.nxx)	120.726	471	53.9%
BFS Crawler	1.785	39	1.41%

The higher precision for Regional Crawler shows more relevant downloaded pages by this crawler. So users with these common interests have found more relevant and updated pages in comparison with using a BFS crawler. In the other words, people in our test, were interested in 53% of retrieved pages.We can conclude that Regional Crawler behaves more intelligently and performs better than a BFS crawler using nxx.nxx weighting scheme.

B. tf-idf vector model results

Giving weights as the sum of term frequencies is too primitive, because it totally ignores important parameters like document frequency (number of documents in which interest term has occurred) and document length. A longer document is more probable to include interest terms, but this does not mean that it is more relevant, so we must prioritize shorter but more relevant documents somehow.

In vector model weighting, a document d_j and region interests I are represented as t-dimensional vectors. The vector model evaluates the degree of similarity of the document d_j with regard to the interests vector I. This similarity can be quantified, for instance, by the cosine of the angle between these two vectors, as is shown in Eq. (1). [18 and 19]

$$sim(dj,q) = \frac{\vec{d}_j \bullet \vec{I}}{\left|\vec{d}_j\right| \times \left|\vec{I}\right|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,I}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,I}^2}} \quad (1)$$

in Eq. (1) $\vec{d}_j$ is the weight vector for document j and $\vec{I}$ is the weight vector for the interests. Since we don't have a fixed document collection we assume the set of downloaded pages from the entire web as our collection. The document collection dynamically changes as the number of crawled pages increase so the weights must be updated after each crawl. In this equation $w_{i,j}$ and $w_{i,I}$ are the weights for term i in document j and in Interests vector I and are calculated as below:

$$w_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}} \times \log\frac{N}{n_i} \quad (2)$$

$$w_{i,I} = (0.5 + \frac{0.5 freq_{i,I}}{\max_l freq_{l,I}}) \times \log\frac{N}{n_i} \quad (3)$$

in Eq.s (2) and (3) N is the total number of downloaded documents and n_i is the number of these documents in which ith interest term appears. $freq_{ij}$ is the frequency of ith interest term in jth document and $\max_l$

$freq_{i,j}$ is the frequency of the term with the maximum number of occurrence in jth document, trivially $\max_i freq_{i,l}$ is the frequency of the most frequent term in the interest vector. The logarithmic expression is called *idf* (inverse document frequency). The best known term-weighting schemes use weights which are given by or by a variation of the above formulas. Such term-weighting strategies are called *tf-idf* schemes [21].

Calculating the similarity between interest vector and documents by dividing them to their lengths is also called "cosine normalization". Cosine normalization reduces the problem of vector component weights for a given document being distorted by a single term; in fact, the normalization factor (vector length) is a function of all the vector components [20]. In this weighting scheme, as was explained the document frequency (idf factor) and document length were parameters we used for weight calculation. Results show how these parameters affect the weighting scheme and crawling path.

Table II shows the results of Regional Crawler extracted pages using this weighting scheme for 330 downloaded pages starting with worldnews index page.

TABLE II

tf-idf RESULTS

URL	http://www.worldnews.com		
Number of Iterations	330		
	Average Weight	Max Weight	Precision
Regional Crawler (tf.idf)	0.283	0.993	9.6%
BFS Crawler	0.315	0.912	1.8%

C. Lnu.ltu term-weighting scheme results

Lnu.ltu [22] is another term-weighting scheme. Lnu is the weighting method for documents and ltu is for query(ies) (interests in our experiments). Therefore, the complete specification of the weighting scheme involves two triples. Each weight is constructed using a term frequency factor (1st letter), a document frequency factor (2nd letter) and a document length factor (3rd letter) [20]. Eq.s (4) to (6) give the formulas necessary for calculating Lnu.ltu weights.

$$l = \ln tf + 1.0 \tag{4}$$

$$L = \frac{1 + \log tf}{1 + \log(average\ \ tf)} \tag{5}$$

$$u = \frac{1}{(slope \bullet \#unique\ \ terms) + (1 - slope) \bullet pivot} \tag{6}$$

n is set to 1.0 because number of documents containing a given term is ignored and t is equal to *idf* factor, described in previous subsection.

In Eq.s (4) to (6) *tf* is the term frequency. *Pivot* and *slope* are parameters of "pivoted unique normalization" [17]. The basic idea behind "pivoted normalization" is that in "cosine normalization" we tend to favor short documents over long ones. In fact, assuming *tf*s be fixed as the length of the documents increase their weight decrease. So there will always be a document length for which probabilities of relevance and retrieval are the same. We call this length the pivot and use a "correction factor" to rotate the old normalization function around pivot so that document lengths below pivot are greater than before and document lengths above pivot are less than before [20]. In "pivoted unique normalization", we use the number of unique terms in the document as the normalization function.

The values for a fixed *pivot* and different *slopes* are given for TREC queries in [17]. We used the same values for web evaluation. The results qualitatively showed that the values also work for web evaluation. Finding the best values for web needs a deeper study on more accurate and variant results from WWW. The values we used for our tests were *pivot=107.89* and *slope=0.2* which seemed reasonable.

Table III shows the results for 500 downloaded pages using Lnu.ltu weighting scheme. Here again the important parameter is the precision calculated manually by several users.

TABLE III

Lnu.ltu RESULTS

URL	http://www.worldnews.com		
Number of Iterations	500		
	Average Weight	Max Weight	Precision
Regional Crawler (Lnu.ltu)	0.445	0.994	28.2%
BFS Crawler	0.188	0.978	3.8%

D. Comparison between evaluation schemes

In a general sense, a crawler may be evaluated on its ability to retrieve "good" pages [12]. But the problem is that how can we recognize good (or relevant) pages. In general the best way to do this is by taking the precision for the downloaded collection. This can be done only on small crawled collections still with a great time expense. Another method is to evaluate crawlers according to the weights they have assigned. Both crawlers assess an equal weight to a specific page because of the same weighting scheme they use. So if a page is weighted higher it shows its superiority to the other page according to weighting parameters (this can be more interest terms or less document length). Although this superiority doesn't mean more relevance, it can be an estimation of

it. In our experiments a higher average weight in most of weighting schemes would result in a higher precision. There still were so many pages with high weight which were not relevant.

Table IV shows a comparison between these three weighting schemes.

TABLE IV

COMPARISION OF DIFFERENT WEIGHTING SCHEMES

	Total Number of Pages	Number of Relevant Pages	Precision
Lnu.ltu	200	40	20%
Vector Space tf.idf	200	32	16%
nxx.nxx	200	6	3%
Normal Crawler	200	0	0%

These results are for 200 pages crawled starting with a seed URL *http://www.spectster.com/* which is a general website containing many fields including news. The Regional Crawler found its way to crawl the Iran news web page after 85 links using Lnu.ltu and tf.idf schemes, but using nxx.nxx it changed its direction before evaluating Iran news web page, so it finally resulted in less relevant pages. The normal crawler didn't happen to crawl news page, let alone Iran news page within 200 cycles.

In all of our experiments with Regional Crawler, using Lnu.ltu weighting scheme we achieved the most precision. But the other two schemes did not get any advantage to each other. In some cases nxx.nxx obtained a greater precision than tf.idf and vice versa. In contrast to previous discussed experiments, in our other test with *http://www.worldnews.com* as the seed URL for 500 page crawls, still Lnu.ltu scheme got the first rank for the number of relevant pages. But nxx.nxx resulted better than tf.idf scheme.

nxx.nxx method is the fastest scheme among all, and the two other weighting schemes generally consumes the same amount of time. They both have to update the whole URL database weights after a new URL is added to the database. Taking into account the problem of performance we can rank weighting schemes this way: Lnu-ltu is the best among all since it best satisfies the need of a focused crawler. nxx-nxx is in the second place since it has a good performance and also does better than tf-idf in some circumstances and tf-idf is the worst since it neither has a good performance nor a guarantee that it gives a high precision.

V. CONCLUSION AND FUTURE WORK

In this Paper we have described Regional Crawler and evaluated its crawling efficiency using different weighting schemes. The main advantage of Regional Crawler over the other kinds of web crawlers is that Regional Crawler updates and locates the new most important pages according to users groups' needs and interests, providing more up-to-date results for the users. The users' interests could be learned from their queries and user's Region could be found from their IPs. The more important a page is for a region's interest, the more frequently it will be looked for, updated and become available. This strategy will help the Search Engines to more closely cater the needs of regions of users.

We have evaluated the Regional Crawler using different weighting schemes and as a Result we have found that Regional Crawler works best by using Lnu-ltu weighting but tf.idf and nxx.nxx have not an obvious advantage to each other -except that nxx-nxx has much a better run time- at least in a limited number of pages in our tests. Early results from our experiments seem very promising, and we intend to widen and deepen our experiments in several dimensions. For example, many improvements can be done on profiling and learning regions. Other interesting things to consider are discovering a shift of interest in some region. For example, a sport event or a political change may spur a lot of request for sites dealing with those issues. These shifts might mean a temporary emphasis on some part of a region's interest or a complete change of interest. An intelligent crawler could detect this shift and respond to it by assigning more weight to those pages and updating them more frequently than usual for a period of time.

VI. REFERENCES

[1] TGIF Google, http://www.stanford.edu/services/websearch/Google/TGIF/outof-index.html, Google at Stanford TGIF presentation 16-May-2003

[2] S. Mayocchi, "A Web Crawler for Automated Location of Genomic Sequences", *Department of Computer Systems and Electrical Engineering, University of Queensland*, BA Thesis, 2001

[3] Internet Metrics and Statistics Guide: Size and Shape, http://www.caslon.com.au/metricsguide13.htm, Version of August 2003

[4] Web Statistics (size and growth), http://wcp.oclc.org/stats.html, 2002

[5] P. Agars, "Architecture of a Search Engine", *An essay submitted to Dublin City University School of Computer Applications for the course CA437*: Multimedia Information Retrieval, 2002

[6] M. Hollander, "Google's Page Rank Algorithm to Better Internet Searching", University of Minnesota, *Morris Computer Science Seminar*, Spring 2003

[7] G. Salton, M.J. McGill, *Introduction to Modern Information Retrieval,* McGraw-Hill, 1983

[8] Sutton R. S., Barto A. G., *Reinforcement Learning: An Introduction,* MIT Press, Cambridge, MA, 1998

[9] Statistics – Web Growth, http://www.upsdell.com/BrowserNews/stat-growth.html, 2003.

[10] T. Suel, V. Shkapenyuk, "Design and Implementation of a High-Performance Distributed Web Crawler", *In Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, San Jose, CA, 2002

[11] Chen, H., Chung, Y., Ramsey, M., and Yang, C. "A Smart Itsy Bitsy Spider for the Web," *Journal of the American Society for Information Science* (49:7), 1998a, pp. 604-618

[12] Menczer F., Pant G., Srinivasan P. and Ruiz M., "Evaluating Topic-Driven Web Crawlers", *In Proceedings of the 24th annual International ACM/SIGIR Conference*, New Orleans, USA, *2001*

[13] S. Brin, L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", *Computer Science Department, Stanford University*, Stanford, USA, 1998

[14] Internet Metrics and Statistics Guide: Size and Shape, http://www.caslon.com.au/metricsguide2.htm, February 2004.

[15] J.J. Rocchio, "Relevance Feedback in Information Retrieval", In G. Salton, *editor The SMART Retrieval System-Experiments in Automatic Document Processing*, Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

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

[17] Singhal, A., Buckley, C., Mandar, M. "Pivoted Document Language Normalization", *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.21-29, 1996.

[18] G. Salton, "The SMART Retrieval System", *Experiments in Automatic Document Processing*, Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[19] G. Salton and M. E. Lesk, "Computer Evaluation of Indexing and Text Processing", *Journal of the ACM*, 15(1):8-36, January 1968.

[20] Ed Greengrass, "Information Retrieval: A survey", DOD Technical Report TR-R52-008-001, 2001

[21] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, "*Modern Information Retrieval*", ACM Press, 1999.

[22] G. Salton, A simple blueprint for automatic boolean query processing, *Information Processing &Management,* Vol. 24, No. 3, pp. 269-280, 1988.

[23] Jason Rennie and Andrew McCallum "Efficient Web Spidering with Reinforcement Learning" , *Proceedings of the 16th international Conference on Machine Learning (ICML-99),* 1999

[24] Junghoo Cho, Hector Garcia-Molina, Lawrence Page "Efficient Crawling Through URL Ordering", *Proceedings of the 7th international World Wide Web Conference,* 1998

[25] Niran Angkawattanawit and Arnon Rungsawang "Learnable Crawling: An Efficient Approach to Topic-Specific web Resource Discovery", *2nd international Symposium on communications and Information Technology (ISCIT 2002),* October 2002

[26] Padmini Srinivasanacd, Joyce Mitchellab, Olivier Bodenreidera "Web Crawling Agents for Retrieving Biomedical Information", *Proceedings of the International Workshop on Bioinformatics and Multi-Agent,* Bologna, Italy, July 15, 2002.

[27] Bin Liu, Chee Wee Chin, Hwee Tou Ng, "Mining Topic-Specific Concepts and Definitions on the web", *Proceedings of the 12th international World Wild Web Conference (www-2003),* Budapest, Hungary, 20-24 May 2003

[28] Soumen Chakrabarti, Martin van den Berg, Byron Domc, "Focused crawling: a new approach to topic-specific Web resource discovery", *Proceedings of the 8th international World Wild Web Conference,* Toronto, Canada, 1999

[29] M. Shokouhi, P. Chubak, "Designing a Regional Crawler for Distributed and Centralized Search Engines", *proceedings of 10th Australian World Wild Web Conference (AusWeb04),* July 2004, Australia

[30] M. Shokouhi, P. Chubak, F. Oroumchian, H. Bashiri, "Designing and Implementation of Regional Crawler As a New Strategy for Crawling the Web.", *proceedings of IADIS International Conference, e-society 2004,* July 2004, Avila, Spain