A Method for Focused Crawling Using Combination of Link Structure and Content Similarity

Mohsen Jamali, Hassan Sayyadi, Babak Bagheri Hariri and Hassan Abolhassani
{m_jamali,sayyadi,hariri}@ce.sharif.edu, abolhassani@sharif.edu
Web Intelligence Research Laboratory,
Computer Engineering Department,
Sharif University Of Technology, Tehran, Iran

***Abstract*— The rapid growth of the world-wide web poses unprecedented scaling challenges for general-purpose crawlers and search engines. A focused crawler aims at selectively seek out pages that are relevant to a pre-defined set of topics. Besides specifying topics by some keywords, it is customary also to use some exemplary documents to compute the similarity of a given web document to the topic. In this paper we introduce a new hybride focused crawler, which uses link structure of documents as well as similarity of pages to the topic to crawl the web**

I. Introduction

The world-wide web, having over 11.5 million pages [1], continues to grow rapidly (according to a Nature magazine article[1], the World Wide Web doubles in size approximately every 8 months). Such growth poses basic limits of scale for today's generic crawlers and search engines. In last years of 90th decade, Alta Vista's crawler, called the Scooter, was running on a 1.5GB memory, 30GB RAID disk, 4x533MHz AlphaServer 4100 5/300 with 1 GB/s I/O bandwidth [2]. In spite of these heroic efforts with high-end multiprocessors and clever crawling software, the largest crawls cover only 30-40% of the web, and refreshes take weeks to a month [3].

The Web in many ways simulates a social network: links do not point to pages at random but reflect the page authors' idea of what other relevant or interesting pages exists. This information can be exploited to collect more on-topic data by intelligently choosing what links to follow and what pages to discard. This process is called *Focused Crawling* [2].

Focused crawling is a relatively new, promising approach for improving the precision and recall of expert search on the Web. In this paper, we describe a crawler that will seek, acquire, index, and maintain pages on a specific topic. Such a focused crawler entails a very small investment in hardware and network resources and yet achieves respectable coverage at a rapid rate, simply because there is relatively little to do. We have selected the topic of *Sports* for our crawler. The crawler starts with a single seed page and tries to fetch the most related pages to *Sports* from the Web. Our crawler maintains the link structure among crawled pages and uses the combination of link structure analysis and similarity of page contents to the topic to rank pages.

[1]http://wwwmetrics.com
[2]http://citeseer.ist.psu.edu/ackerman97learning.html
[3]http://citeseer.ist.psu.edu/lawrence98searching.html

The rest of the paper is organized as follows. Section 2 reviews and discusses on related works. In Section 3 we illustrate our algorithm for focused crawling ,then in section 4 the metric for content similarity measurement will be described in detail. Our evaluation methods and experimental results for real data are described in Section 5. Finally there is a conclusions and discussions on future works in section 6.

II. Related Works

In some early works on the subject of focused collection of data from the web, web crawling was simulated by a *group of fish* migrating on the Web [3]. In the so called *fish search*, each url corresponds to a fish whose survivability is dependant on visited page relevance and remote server speed. Page relevance is estimated using a binary classification by using a simple keyword or regular expression match. Only when fish traverse a specified amount of irrelevant pages they die off. The fish consequently migrate in the general direction of relevant pages which are then presented as results.

[4] considers an ontology-based algorithm for page relevance computation. After preprocessing, entities (words occurring in the ontology) are extracted from the page and counted. Relevance of the page with regard to user selected entities of interest is then computed by using several measures on ontology graph (e.g. direct match, taxonomic and more complex relationships). The harvest rate is improved compared to the baseline focused crawler (that decides on page relevance by a simple binary keyword match).

[5] uses relevance feedback to predict page quality. They also made separate use of relevance feedback in scoring topic relevance for evaluation purposes only: quality *RF* and relevance *RF*. Both use the term selection methods to identify extra query words and phrases. So they developed a classifier for predicting the relevance of a link target, based on features in the link's source page. They evaluated a number of learning algorithms provided by the *Weka* package, such as k-nearest neighbor, Naive Bayes, and C4.5. Since then they also evaluated Perception. The C4.5 decision tree was the best amongst those evaluated. The classifier is based on words in the anchor text, words in the target url and words in the 50 characters before and after the link (link context).

III. Proposed Focused Crawling Algorithm

In Web, ordinary hyperlinks in pages are not randomly created but in most cases represent the author's view about other pages. Also the contents of pages are another source to relate them to a domain (e.g. Sports in our work). In this article a crawler which uses a combination of links structure and contents to do focused crawling is introduced. To implement it we need to maintain link structure of pages and also introduce a metric for measuring the similarity of a page to a domain. Later one is explained in detail in section 4. Assuming having it, in this section we explain the crawling algorithm.

A. The crawler architecture and algorithm

Figure 1 shows the architecture of the crawler. Initially a single page is considered as primary seed. An ID is assigned to it and together with its url is stored in the database (i.e. in a table named 'Seed Pages'). On addition of a page to

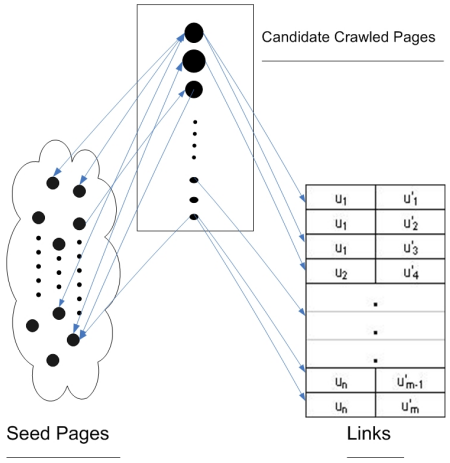

Fig. 1. Architecture of Our focused crawler

seed pages, following tasks are done (Seed is empty at start, and an initial seed page would be added to it at beginning. Then in each step one of the fetched pages will be added to the seed. Method for selecting such a fetch page comes in following). Each seed page has some links to the other pages. First, such links are downloaded and stored in a special folder, i.e. download folder. Then address of the page and some of its attributes is stored in the database. Attributes include similarity degree of the page to the domain (Sports), number of links from the page to seed pages and number of links from seed pages to it.

The similarity degree is a number between 0 and 1 and as mentioned its computation is discussed in section 4. To compute number of links from seed pages to the downloaded page two possibilities exists:

- If this page was not downloaded before then it was not stored in database as 'Candidate Crawled Page' and then it would be stored in Candidate Crawled Page. It is obvious that number of links from seed pages to this page is 1 (since the page is downloaded in this stage and there was no other seed pages linking to it in database).
- If the page was downloaded before it means it is in database then we only need to increase links from seed pages by 1.

To compute number of links from the crawled page to seed pages all of its links is extracted and compared with seed pages in the database. To prevent re-extraction of links in crawled pages, they are stored in a separate table (Links table in the figure 1). Now pages in Candidate Crawled Pages table is ordered based on a metric and in each stage a page with the highest rank is selected and added to Seed Pages. The metric is based on a combination of three values stored for each Candidate Crawled Page as below:

$$Rank(p) = (links\_to\_seed(p) + links\_from\_seed(p)) \times (0.1 + content\_similarity(p)) \quad (1)$$

In this formula p represents a page from Candidate Crawled Pages and $Rank(p)$ is its rank. $links\_to\_seed$ and $links\_from\_seeds$ are number of links from the page to seed pages and number of links from seed pages to it, respectively. Also $content\_similarity$ is similarity value of the page and the domain (explained in detail later in section 4). To care about pages with zero content similarity to the domain but with high number of links to and from seed pages (like images and Flash animations) a constant value (0.1) is added to $content\_similarity$ value. 0.1 is selected to have this effect that a page with complete similarity to the domain has 11 times higher value than a page with zero similarity (experiments have shown satisfactory results for this value). After moving a candidate page with highest rank to seed pages it is needed to adjust links values for the remaining candidates (it is done by a few simple query and update operations).

In the next step we repeat the same operations for new pages in Seed Pages. The experimental results shows that the time needed for this algorithm is not higher than an ordinary crawler (e.g. BFS Crawler). It is almost 1.3 times of an ordinary crawler and considering higher precision of the algorithm and the less amount of resources it needs it seems an acceptable performance.

IV. Content Similarity Measurement

Content Similarity: $String \times String \rightarrow [0 \dots 1]$ returns the degree of similarity of a page to a domain, based on the keywords and phrases used in that domain. In its ideal case this function should have the property that if its value for x_1 is greater than for x_2 then we can conclude that the similarity of x_1 to the domain is higher than x_2. The main idea behind this function is that usually similar words and phrases are used in pages belonging to a specific domain.

A. Details of proposed algorithm

For a given domain, we first apply a clustering algorithm to create groups of documents related to the domain, each represents a sub-domain. In this research we used Vivisimo[4]

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

but it is also possible to first crawl pages and create sub-domains based on them. For each sub-domain, a vocabulary is created by selecting a number of pages (randomly) from the respective cluster and select terms of them having occurrences above a specified threshold *minOccurence*(we used 2 as this threshold value). Having vocabulary vectors for sub-domains,

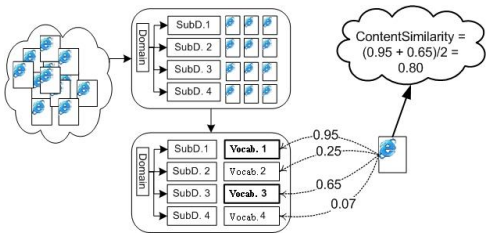

Fig. 2. Calculating Content Similarity of Web pages

a given page is compared with them and content similarity for the k-most similar vectors to the page by averaging similarities is computed. In the experiments, $k = 2$ has shown good results. For $k = 1$ our experiments does not show promising result, that might be because of noise in vocabulary vectors. Our proposed metric is shown in formula 2 in which $\max_k$ shows a set which contains k greatest values of the set and v_i shows the vocabulary of the i_{th} sub-domain.

$$ContentSimilarity = Average(\max_k \frac{|v_i \cap s|}{|s|}) \quad (2)$$

V. Evaluation and Experimental Results

There are many indicators of the performance of a focused crawler. Relevance (precision), coverage (recall) and quality of resource discovery are some of them. We will measure precision and will discuss on the quality of resource discovery. It is extremely difficult to measure or even define recall for a focused crawler, because we have a rather incomplete and subjective notion of what is 'good coverage' on a domain.

The primary metric which was used to evaluate the performance of the crawling system was the harvest rate P(C), which is the percentage of the web pages crawled which are related to the domain. Most focused crawlers have used this metric [6],[7],[4]. The core improvement of our focused crawler derives from combining link structure analysis and content similarity. We therefore compare the efficiency of our focused crawler with a usual breadth-first crawler, using harvest rate metric.

We've ran our focused crawler two times. In each time there was an initial seed page. In the first run the seed page was a good hub[5] for sport domain (Search result page of Google for keyword *Sport*). The second run was initiated with a more usual page (www.yahoo.com). Both two runs of our crawler are compared with an ordinary BFS crawler.

25000 pages were fetched with crawler in each time of run. We've assigned ID for each page. Results for running the

[5]A hub is page having many links to authority pages

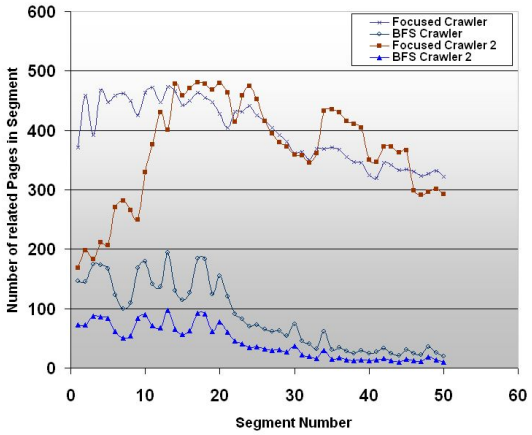

Fig. 3. Related pages in each segment(500 pages) of crawled(fetched) pages. In first time of running, focused crawler works nearly 2.5 times better at the start, and as the number of crawled pages increases this ratio also increases so much (focused crawler works nearly 16 times better than a usual BFS crawler at the end of crawling). When we have Yahoo as the initial seed page, focused crawler works two times better at the start, and as number of crawled pages increases, this ratio will increase. This time, the number of related documents are low for focused crawler, but it increases steeply and reaches a maximum and after that the number of related pages will decrease

crawler in both times of running are shown in Figures 3 and 4(Curves for the second run of the crawlers are labeled with postfix 2 in diagrams -Focused Crawler 2, BFS Crawler 2-). In the figure 3, number of related pages in each segment is shown. A segment in this figure represents 500 pages, meaning that each 500 crawled pages(in chronological order) for a segment. As figure 3 shows, in first run of focused crawler, the number of related pages in first two segments is not so high; it is meaningful since in primary steps of crawling algorithm there is not so much page in Candidate Crawled Pages table and so there is some noise. But as number of crawled pages increases, number of related pages will be higher and after some steps this amount gradually will decrease because of getting farther from the seed pages. Figure 4 also shows the harvest rate of the focused crawler. There are some noise at start, because of the outlinks of the initial seed page which may not be related to the relevant documents. But as the number of crawled pages increases, the chart would be smoother and harvest rate increases and reaches a maximum, then the harvest rate decreases. Result for running the crawler for the second time (with www.yahoo.com as its initial seed page) are also shown in Figure 3,4. In figure 3 we can see that the number of related pages is low at start(less than 200 pages are related in first segment), it's because the initial seed page does not relate to our domain, and so the ratio of related pages is very low. But as the number of fetched pages increases, this ratio will increase rapidly and will reach a maximum and then decreases. this decrease is not smooth and has local maximums. We think it's just because of this matter that initial seed page doesn't relate to domain and this causes noises. Figure 4 shows the harvest rate of running the focused crawler for the second time. The harvest rate is very low at start but it increases rapidly

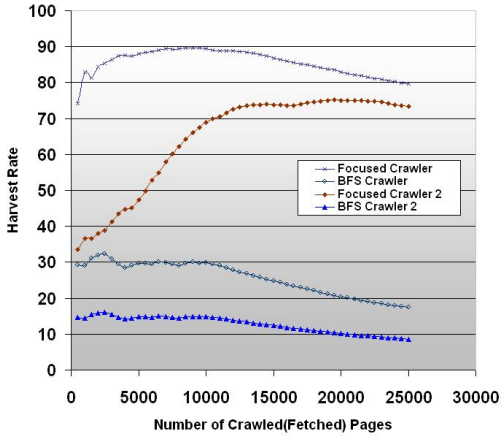

Fig. 4. Harvest rate of the crawled pages. In the first run, the harvest rate is near 90% in focused crawler, but less than 30% in BFS crawler. Harvest rate decreases as number of crawled pages increases. Curves reporting results for the second run of crawlers show that the harvest rate is near 33% in focused crawler at start, the rate would increase and reach a maximum value of 75% and after that it will smoothly decrease, but in a BFS crawler harvest rate is less than 15% and would decrease linearly

and after some time it smooths.

Comparing results for both runs of the focused crawler shows that its harvest rate is obviously better than a BFS crawler (ratio of 16 for the first run and 30 for the second). Of course time consumed by our crawler is more than an ordinary crawler(1.3 times more) but the harvest rate gained by our crawler is so high that encourages its usage. Summarizing all the experiments, we have experienced 4 crawlers : two focused crawlers with different pages as initial page, and two BFS crawlers. Table I shows some statistical reports of the crawlers.

We also ran our algorithm for 3 different topics. Figure 5 compares the harvest rate for these topics. As can be seen from the chart there is no great differences between them.

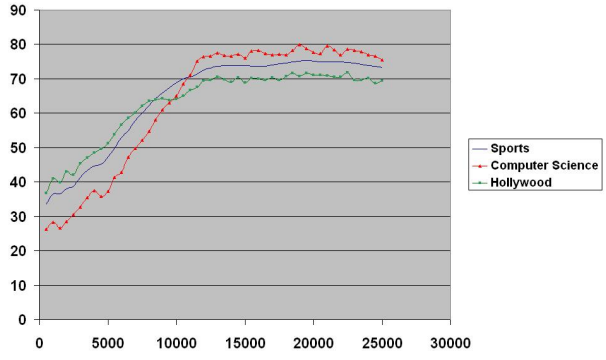

Fig. 5. Harvest rate of the crawled pages for different topics.

VI. Conclusions

The motivation for focused crawling comes from the poor performance of general-purpose search engines, which depend on the results of generic web crawlers. The focused crawler is a system that learns the specialization from examples, and then explores the web, guided by a relevance and popularity rating mechanism. It filters at the data-acquisition level, rather than as a post-processing step.

Crawler Type	SeedPage	TCP	RPC	RCT	AHR(%)
Focused	Search Result of Google for Sport	25000	19904	1.3	85.5
BFS	Search Result of Google for Sport	25000	4346	1	25.56
Focused	www.yahoo.com	25000	18320	1.32	67.2
BFS	www.yahoo.com	25000	2159	1.05	12.72

TABLE I

Comparing Crawlers with respect to their harvest rate. (*TCP*: Total Crawled Pages, *RPC*: Related Pages' Count, *RCT*: Relative Crawling Time, *AHR*: Average Harvest Rate), *AHR* is the mean of harvest rates in each segment

In this paper, we have introduced a simple framework for focused crawling using combination of two existing methods, the Link Structure analysis and Content Similarity. Our generic framework is more powerful and flexible than previously known focused crawlers.

In experimental evaluations, we have compared our crawler with the unfocused one. For this purpose we have studied their behavior in two different tests, one uses a hub as seed page, and the other uses a non-related page. The proposed method shows superority over non-focused one with a high harvest-rate.

VII. Acknowledgments

This research was in part supported by a grant from the Institute for Studies in Theoretical Physics and Mathematics (I.P.M.).

References

[1] A.Gulli and A.Signorini, "The indexable web is more than 11.5 billion pages." Chiba, Japan: 14th international conference on World Wide Web, 2005, pp. 902–903.

[2] B. Novak, "A survey of focused web crawling algorithms." Ljubljana, Slovenia: SIKDD at multiconference IS, Octobr 2004. [Online]. Available: http://eprints.pascal-network.org/archive/00000738/01/BlazNovak-FocusedCrawling.pdf

[3] P. M. E. De Bra and R. D. J. Post, "Information retrieval in the World-Wide Web: Making client-based searching feasible," *Computer Networks and ISDN Systems*, vol. 27, no. 2, pp. 183–192, 1994. [Online]. Available: citeseer.ist.psu.edu/debra94information.html

[4] M. Ehrig and A. Maedche, "Ontology-focused crawling of web documents," in *Proc. of the Symposium on Applied Computing, March, Florida, USA*, 2003.

[5] T. Tang, D. Hawking, N. Craswell, and K. Griffiths, "Focused crawling for both topical relevance and quality of medical information," in *Proceedings of CIKM'2005*, Bremen, Germany, 2005.

[6] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu, "Intelligent crawling on the world wide web with arbitrary predicates," in *World Wide Web*, 2001, pp. 96–105. [Online]. Available: citeseer.ist.psu.edu/aggarwal01intelligent.html

[7] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: a new approach to topic-specific Web resource discovery," *Computer Networks (Amsterdam, Netherlands)*, vol. 31, pp. 1623–1640, 1999. [Online]. Available: citeseer.ist.psu.edu/chakrabarti99focused.html