A Community Base Algorithm for Discovering Web Crawler Seeds Set

Shervin Daneshpajouh

Computer Engineering Department
Sharif University of Technology
daneshpajouh@ce.sharif.edu

Joint work with
Mojataba Mohammadi Nasiri
and
Mohammad Ghodsi

Sharif University of Technology

Nowadays web has an imperative impact on our daily life providing required information. Regarding to [1] size of web is estimated 11.5 billion pages at 2005. This size is now even larger and become larger as the time elapse. Web search engine like Google, Yahoo, MSN,… has an important role for facilitating information access. Because of huge amount of information in web, without these search engine people can not find their relevant information from billion's pages in web. A web search engine consists of three main parts: A crawler that retrieves web pages, an indexer that builds indexes, and a searcher. Figure 1 shows anatomy of a large scale web search engine [2]. A major question a crawler has to face is which pages to retrieve so as to have the "most suitable" pages in the collection [3]. Crawlers normally retrieve a limited number of pages. In this regard the question is how fast a crawler collects the "most suitable" pages. A unique solution to this question is not likely to exist. In what follows, we try to answer this question.

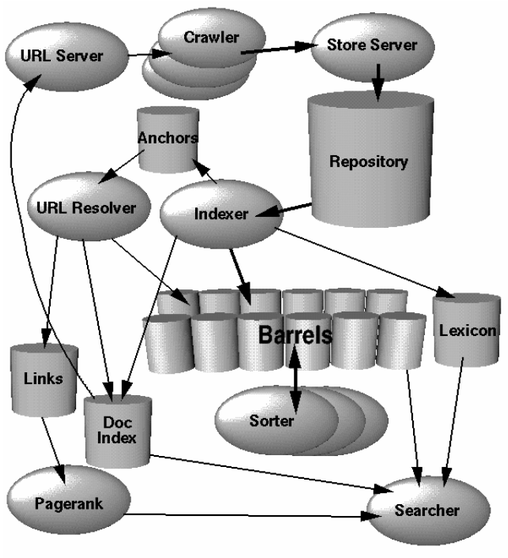

Figure 1. Anatomy of a large scale web search engine, [2]

Different algorithms with different metrics have been suggested to lead a crawl towards high quality pages [4,5]. In [4] Cho, Garcia-Molina, and Page suggested using connectivity-based metrics to do so. To direct a crawl, they have used different ordering metrics: breadth-first, and backlink count, and PageRank and random. They have revealed that performing a crawl in breadth-first order works nearly well if "most suitable" pages are defined to be pages with high PageRanks.

Najork and Wiener extend the result of Cho, Garcia-Molina, and Page examined the average page quality over time of pages downloaded during a web crawl of 328 million unique pages. They have showed that traversing the web graph in breadth-first search order is a good crawling strategy.

Regarding to Henzinger's work [3] better understanding of graph structure might lead to a more efficient way of the web crawling. We use this idea in to develop our algorithm. First we define the "most suitable" pages. We then show how a crawler can retrieve most suitable pages. We use three metrics to measure the quality of a page. The first metric is community of pages. A collection of good crawls should contain pages from different communities. The second metric is PageRank [2] of a page. Pages with high PageRank are the most important pages in web. The third metric is number of visited pages at iterations. A good crawler will visit more pages in less iteration.

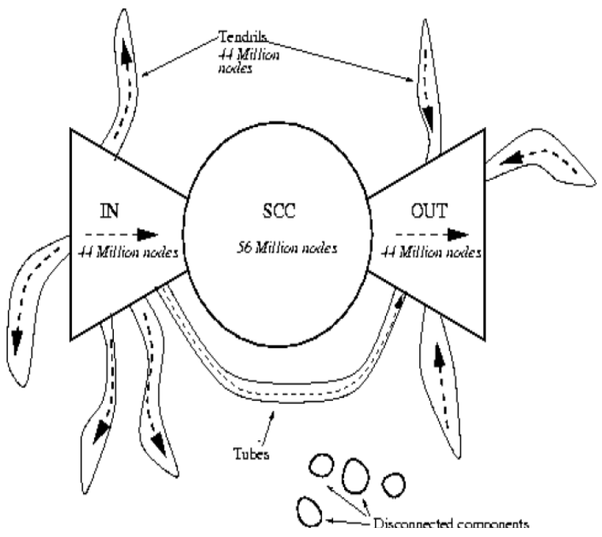

Figure 2. Connectivity of The Web

We present an algorithm for extracting seed set from a previously crawled pages. Using offered metrics we show that starting extracted seeds by our algorithm a crawler will quickly collect most suitable pages.

We have studied different community extraction algorithm: PageRank, Trawling, HITS, and Network flow base community discovery. Regarding our analysis we use HITS ranking without keyword search in our algorithm for community discovery and collecting seeds set. We have found bipartite cores very useful for selecting seeds set. Bipartite cores contain Hub and Authority pages. Since we are interested in having Authority pages in our crawl, we would need to start crawling from Hub pages. Hub pages are durable pages, so we can count on them for crawling.

In comparison with Trawling algorithm, HITS ranking finds bipartite cores more quickly. Network flow base community discovery algorithm searches for dense sub graphs and finds communities containing bipartite cores. Yet, it does not provide information about the value of the pages in the extracted community. As HITS algorithm ranks web pages, its ranking can be used for evaluating the importance of selected pages in bipartite cores.

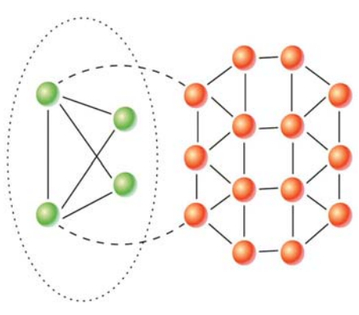

Figure 3. Relation Between Community Extracted by NetworkFlow and Its relation to HITS Bipartite Core [10]

From the structure presented by [6] and their analysis the most important web pages in the web are expected to be in SCC+OUT. Figure 2 shows the structure of web. From [2] we know that nodes or web pages with high page rank are the most valuable pages in the web. In addition from [3, 7, 8] we understand that bipartite cores are one of the valuable sources in he web usually called hubs and authorities. Besides we know that web contains thousands of different communities. In [7] Kleinberg used keyword and a ranking method for finding hub and authorities pages which result contains pages in one or more communities. In [9] Flake et al suggested a method based on network flow algorithm for finding web communities. The relation between communities extracted from their algorithm and Kleinberg's community is that result of Kleinberg's algorithm is expected to be a subset of Flake et al algorithm [10]. See Figure 3 for a sample of relation between results of these two algorithms.

A crawler normally do not crawl the entire web and continue to retrieve a limited number of pages. Crawlers tend to collect "most suitable" pages of web rapidly. We defined "most suitable" pages of web are pages with high PageRank. These pages are pages that HITS algorithm calls them Authority pages. The difference is that HITS algorithm finds the authorities pages relating to key word but PageRank shows the importance of a page in the whole web. As well we know that good hubs links to good authorities. If we be able to extract good hubs from a web graph and from different communities than we will be able to download good authorities which have high PageRank of different communities.

We use HITS ranking without keyword search on previously crawled web pages and than prune the resulted sub graph than we run again HITS ranking and prune it and repeat respectively. We show that selected hub nodes from the resulted sub graph of iterations can solve crawl problem mentioned in this paper.

We assume that we have a web graph of crawled web pages. Our aim is to extract seeds set from this graph so that a crawler can collect the most important pages of web in less iteration. To do this we run HITS ranking algorithm on this graph. This is the second step of HITS algorithm. In the first step it searches the keyword in an index-base search engine. For our purpose we ignore this step and only run the ranking step on the whole graph. In this way, bipartite cores with high Hub and Authority rank will become visible in the graph. Then we select the most highly ranked bipartite core using two algorithms we suggest, namely: extracting seeds with fixed size, extracting seeds with fixed density. Then we remove this sub-graph from the graph and repeat ranking, seed extraction and sub-graph removal steps till we have enough seeds set.

A question that may arise is that when repeating the step why we run HITS ranking again? Isn't one time ranking enough for whole steps? The answer is removing bipartite core in each step modify the web graph structure we are working on. In this regard, re-ranking change the hub and authority rank of bipartite cores in web graph. Removing high ranked bipartite core and re-ranking web graph drive appeared bipartite cores be from different communities. Thus, a crawler will be able to download pages from different communities starting these seeds.

We have experimented our algorithm using web graph of UK 2002 containing 18,520,486 nodes and 298,113,762 edges, and UK 2005 containing 39,459,925 nodes and 936,364,282 edges [11,12,13,14]. Our experiments prove that extracted bipartite cores have a reasonable distance from each other (See figure 6).

The other question that may arise is that if a crawler starts using seeds resulted from our algorithm, why would the results of crawl lead to the most suitable pages. The answer is that in iterations of algorithm we select and extract high ranked bipartite cores from web graph. Extracted bipartite cores have high hub or authority rank. It is expected that pages with high hub rank link to pages with high PageRank. Our experiments on UK 2002 and UK 2005 prove the correctness of this hypothesis.

We have compared the result of the crawls starting from extracted seeds set produced by our algorithm, with crawls starting random nodes. Consequently, our experiments shows that the crawl staring from seeds set identified by our algorithm find most suitable pages of web very faster in comparison with a random crawler (See figure 7 and 8).

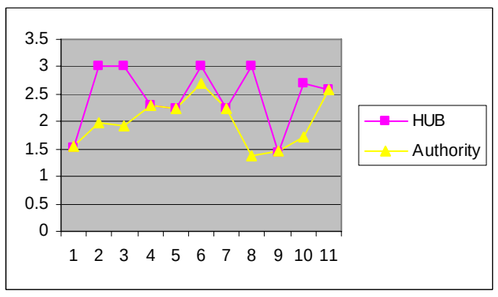

Figure 4. Log-Log diagram of Hub and Authority sizes Extracted from UK 2002

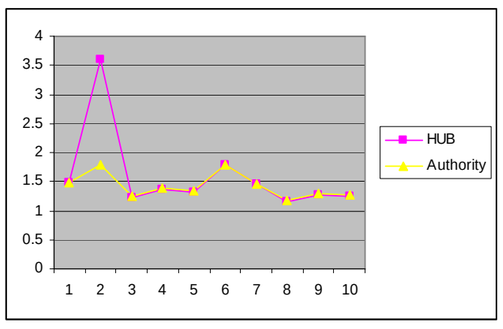

Figure 5. Log-Log diagram of Hub and Authority sizes Extracted from UK 2005

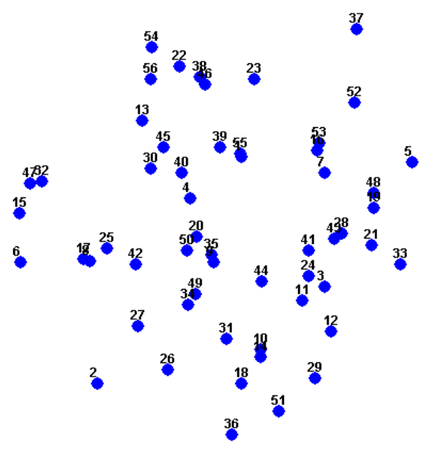

Figure 6. Graphical Presentation of Distances between 56 extracted seeds from UK-2002 by our algorithm. Numbers besides of each node indicate the iteration number, in which related node has been extracted.

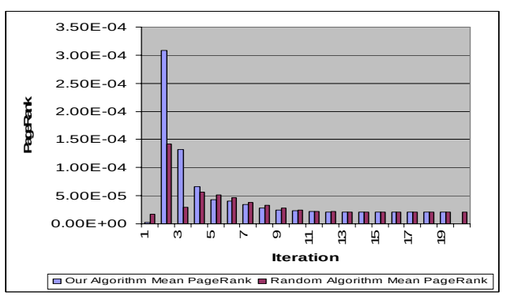

Figure 7. Comparison of PageRank of Crawled pages starting 10 seeds extracted by our method on UK-2002 V.S Random Seeds

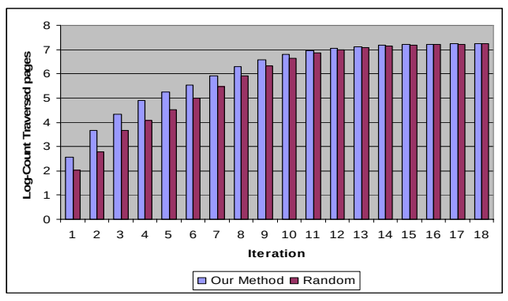

Figure 8. Comparison Log Count diagram of pages visited at each Iteration starting 10 seeds extracted by our method V.S 10 seeds selected randomly from uk-2002

REFERENCES

[1] Gulli, A., and Signorini, A. The Indexable Web is More than 11.5 billion pages. *WWW (Special interest and tracks and posters)*, (May. 2005), 902-903.

[2] Brin , S. and Page, L. *The anatomy of a large-scale hypertextual Web search engine.* Proceedings of the seventh international conference on World Wide Web 7, Brisbane, Australia, 1998, 107 – 117.

[3] Henzinger, M. R. Algorithmic challenges in Web Search Engines. *Internet Mathematics*,Volume 1, Number 1, 2003, 115-123.

[4] Cho,J. Garcia-Molina, H. and Page, L. *Efficient Crawling through URL ordering.* In Proceedings of the 7th International World Wide Web Conference, pages 161-172, Brisbane, Australia, April 1998. Elsevier Science

[5] Najork, Wiener, J. L. *Breadth-First Search Crawling Yields High-Quality Pages*, Proceedings of the 10th international conference on World Wide Web WWW '01, 2001.

[6] Andrei Z. Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, Janet L. Wiener: *Graph structure in the Web*. Computer Networks 33(1-6): 309-320 (2000)

[7] Jon M. Kleinberg*: Authoritative Sources in a Hyperlinked Environment*. J. ACM 46(5): 604-632 (1999)

[8] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins: *Trawling the Web for Emerging Cyber-Communities*. Computer Networks 31(11-16): 1481-1493 (1999)

[9] Gary William Flake, Steve Lawrence, C. Lee Giles, Frans Coetzee: *Self-Organization and Identification of Web Communities*. IEEE Computer 35(3): 66-71 (2002)

[10] J. Kleinberg, S. Lawrence. *The Structure of the Web*. Science 294(2001), 1849.

[11] Laboratory for Web Algorithmics, http://law.dsi.unimi.it/

[12] Paolo Boldi and Sebastiano Vigna, The WebGraph framework I: Compression techniques. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 595-601, Manhattan, USA, 2004. ACM Press.

[13] Boldi, P., Codenotti, B., Santini, M., Vigna, S., UbiCrawler: A Scalable Fully Distributed Web Crawler, Journal of Software: Practice & Experience, 2004, volume 34, number 8, pages 711—726.

[14] Albert, R. Jeong, H. Barabasi, A.L. *A random Graph Model for massive graphs*, ACM symposium on the Theory and computing 2000.