Effective Web-Scale Crawling Through Website Analysis

Iván González[*]
Carnegie Mellon University
School of Computer Science
Pittsburgh, Pennsylvania
ieg@cs.cmu.edu

Adam Marcus[*]
Rensselaer Polytechnic Institute
Computer Science Department
Troy, New York
marcua@cs.rpi.edu

Daniel N. Meredith,
Linda A. Nguyen
IBM Almaden Research Center
San Jose, California
{dnm, lan}@us.ibm.com

ABSTRACT

The web crawler space is often delimited into two general areas: full-web crawling and focused crawling. We present netSifter, a crawler system which integrates features from these two areas to provide an effective mechanism for web-scale crawling. netSifter utilizes a combination of page-level analytics and heuristics which are applied to a sample of web pages from a given website. These algorithms score individual web pages to determine the general utility of the overall website. In doing so, netSifter can formulate an in-depth opinion of a website (and the entirety of its web pages) with a relative minimum of work. netSifter is then able to bias the future efforts of its crawl towards higher quality websites, and away from the myriad of low quality websites and crawler traps that litter the World Wide Web.

Categories and Subject Descriptors

D.2.11 [**Software**]: Software Architecture; H.2 [**Information Systems**]: Information Storage and Retrieval

General Terms

Performance, Design

Keywords

WebFountain, UIMA, netSifter, crawling, sampling

1. INTRODUCTION

There are numerous challenges facing today's web crawlers. The Internet has reached such proportions that a crawler can neither be expected to scan the Web in its entirety, nor refresh all content in a timely manner. Content of questionable merit has proliferated, and in an effort to conserve constrained resources such as bandwidth, processing time and storage, a crawler must avoid such content while directing its efforts toward the discovery of higher-value content, and refreshing known good content.

We implemented netSifter within IBM's WebFountain [3] to counter the challenges faced by current crawler methodologies. netSifter is a scalable, flexible architecture that approaches the Web as a collection of websites, so as to avoid evaluating web pages one by one. The unique concept of the solution is to rank the URL frontier by making page-level judgements based on knowledge of the originating site. This site knowledge is created by performing various analyses on a sample of pages from the site, and subsequently formulating an overall site score. Future pages from the site can then be prioritized in relation to other pages via the site score. The rationale for this idea is that pages from a common site are related, and share characteristics indicative of quality.

netSifter is extensible to varied needs, allowing it to take advantage of both modern focused crawling techiques [1] and full-web crawling strategies [2]. For example, netSifter can make use of relatively expensive content-based analyses while remaining scalable, since only a sample of pages from a site are examined. Additionally, netSifter can make simplified use of popularity measures by, for example, examining outlinks from the sample set of web pages, or by incorporating link-based ranking results into the site scores periodically. Since netSifter employs a plurality of analysis techniques, a website is not excluded or included based on any one metric.

2. SYSTEM DETAILS

2.1 Architecture

The architecture is divided into two stages: online and offline analysis. The online analysis stage is comprised of the scheduler and the online analysis manager. These components direct the crawl, and route pages to offline analysis. The offline analysis stage contains a UIMA [4] analytics chain which performs more extensive content inspection and analysis in order to generate site scores. This analytics chain can be distributed across many nodes via a service-oriented architecture, and is able to run independently and asychronously with respect to the main crawling process. The individual components are described below.

- *Site Score Database* The site score database serves as the link between the online and offline analysis stages, where online analysis queries for site scores in order to prioritize the URL frontier, and offline analysis generates or updates them. Site scores represent the relative value of a site compared to any other site.
- *Scheduler* The crawler perpetually iterates over a list of all URLs that have been discovered or seeded. The list is retrieved in batches, and each URL in the batch is given a score based on its website's score. The batch

[*] Work done at IBM Almaden Research Center.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-332-9/06/0005.

	Pages	Sites	ODP (Sites)	ODP (Pages)	OPD Utility	ODP Utility (Weighted)	SiteRank Utility
unbiased	8,290,541	95,298	35.8%	60.7%	5,423,823	1,564,620,987	251.9
netSifter	7,245,369	65,747	43.0%	67.7%	5,075,633	4,364,345,087	438.1

Table 1: Results for unbiased and netSifter crawls

is sorted by score in descending order, and then sent to the fetcher. The fetcher is allocated a time period in which to fetch as many URLs as possible, though it is generally not able to exhaust a given list. If a site score does not exist, a slightly higher than neutral score is assigned, as unscored sites are favored for their potential to contain novel content.

- *Online Analysis Manager* In addition to being sent to the main system for storage and indexing, all newly fetched pages are routed to the online analysis manager, which determines whether or not a page should be sent to the offline analysis stage. If a page is empty, contains any "soft" errors, or fails other basic data-validity tests, it is discarded. A page is sent offline if no site score for the page exists, or if a sufficient period of time has elapsed since a site score was produced.

- *Offline Analytics Chain* The offline analytics chain passes a page through various annotators, each of which scores the page based on various algorithms. The final annotator in the chain combines the individual scores into a weighted average and applies optional heuristics. The final page score is then stored temporarily in a database. Once a sufficient sample of pages for a site have been collected, those page scores are aggregated and submitted to the site score database.

2.2 Sampling Method

netSifter employs a sampling method to determine that a sufficient sample has been collected from a site. It is assumed that the sample a crawler generates consists of independent and identically distributed pages. A sufficient sample is defined as being a sample which produces a page-score mean $\bar{x}$ which represents the population mean μ within $\mu \pm \delta$ at a confidence level of 95% using a standard t-test.

We require a website to pass the t-test for three sequential observations in order to provide an opportunity for the crawl to move past locally consistent content, and find a better estimate of the mean for the entire site. A maximum sample size is set to avoid analyzing too many pages for websites which are unable to produce a consistent site score within a reasonable sample size.

3. RESULTS AND CONCLUSION

To validate that netSifter accurately measures site quality, we compared website scores to SiteRank (a site-level variant of PageRank) scores. A list of corresponding netSifter and SiteRank scores for sites was produced, and sorted in descending order of SiteRank score. Sites were grouped into buckets, and counts of netSifter scores greater than, equal to, and less than 0 were generated. The results can be seen in Figure 1. There is a positive correlation between higher netSifter scores and site-connectedness.

Among the top 1000 SiteRank scores, 148 sites were scored negatively by netSifter. There were many Asian language, spam, link farm, and adult content sites. The presence of Asian sites indicates that some annotators improperly handle non-English content. netSifter correctly scored spam and adult content sites, even though these sites were rated well by SiteRank. Among the bottom 1000 SiteRank scores, 373 sites were scored positively by netSifter. Manual examination of these sites rated a large majority as postive. This shows that netSifter was able to identify interesting sites which were not well-connected.

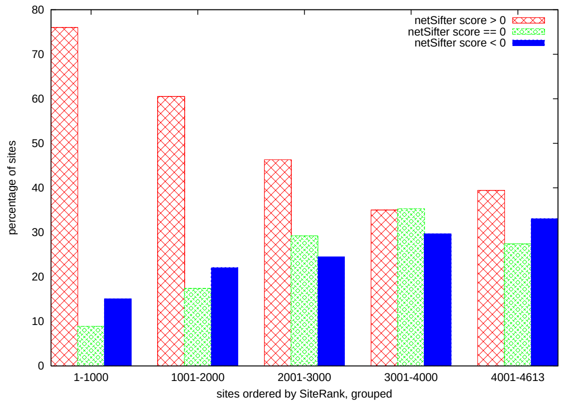

Figure 1: netSifter and SiteRank score correlation

The second experiment compares a netSifter crawler against an unbiased crawler. The utility of the crawlers was measured using *ODP Utility* (the count of pages crawled from sites which appear in the ODP listings, *ODP Utility (Weighted)* (the sum of the number of pages crawled from a given site multiplied by the number of site appearances in the ODP, and *SiteRank Utility* (the number of pages crawled from a given site multipled by the SiteRank of that site. The results are shown in Table 1. Though netSifter crawled fewer pages than the unbiased crawl, it outperformed it on *ODP Utility (Weighted)* and *SiteRank Utility*.

netSifter demonstrates that by exploiting the logical association of a web page to a website, and then forming an estimate of the overall quality of a website, the URL frontier of a web-scale crawler can be effectively prioritized to bias the crawler towards higher-quality content.

4. REFERENCES

[1] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 1999.

[2] J. Cho, H. García-Molina, and L. Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 1998.

[3] D. Gruhl, L. Chavet, D. Gibson, J. Meyer, P. Pattanayak, A. Tomkins, and J. Zien. How to build a webfountain: An architecture for very large-scale text analytics. *IBM Systems Journal*, 43(1):64–77, 2004.

[4] International Business Machines Company. `http://www.research.ibm.com/UIMA`.