Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery[1]

Soumen Chakrabarti[2]
Indian Institute of Technology
Bombay 400 076
soumen@cse.iitb.ernet.in

Martin van den Berg[3]
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg 4
Palo Alto, CA 94304
vdberg@pal.xerox.com

Byron Dom
IBM Almaden Research Center
650 Harry Rd
San Jose, CA 95120
dom@almaden.ibm.com

Abstract

The rapid growth of the world-wide web poses unprecedented scaling challenges for general-purpose crawlers and search engines. In this paper we describe a new hypertext resource discovery system called a *Focused Crawler*. The goal of a focused crawler is to selectively seek out pages that are relevant to a pre-defined set of *topics*. The topics are specified not using keywords, but using exemplary documents. Rather than collecting and indexing all accessible web documents to be able to answer all possible ad-hoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the web. This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date.

To achieve such goal-directed crawling, we design two hypertext mining programs that guide our crawler: a *classifier* that evaluates the relevance of a hypertext document with respect to the focus topics, and a *distiller* that identifies hypertext nodes that are great access points to many relevant pages within a few links. We report on extensive focused-crawling experiments using several topics at different levels of specificity. Focused crawling acquires relevant pages steadily while standard crawling quickly loses its way, even though they are started from the same root set. Focused crawling is robust against large perturbations in the starting set of URLs. It discovers largely overlapping sets of resources in spite of these perturbations. It is also capable of exploring out and discovering valuable resources that are dozens of links away from the start set, while carefully pruning the millions of pages that may lie within this same radius. Our anecdotes suggest that focused crawling is very effective for building high-quality collections of web documents on specific topics, using modest desktop hardware.

Keywords: Web resource discovery, classification, categorization, topic distillation.

1 Introduction

The world-wide web, having over 350 million pages, continues to grow rapidly at a million pages per day [2]. About 600 GB of text changes every month [19]. Such growth and flux poses basic limits of scale for today's generic crawlers and search engines. At the time of writing[4], Alta Vista's crawler, called the *Scooter*, runs on a 1.5 GB memory, 30 GB RAID disk, 4×533MHz AlphaServer 4100-5/300 with 1 GB/s I/O bandwidth. Scooter connects to the indexing engine *Vista*, which is a 2 GB memory, 180 GB RAID disk, 2×533MHz AlphaServer 4100-5/300. (The query engine is even more impressive, but is not relevant to our discussion.) Other giant web crawlers use similar fire-power, although in somewhat different forms, e.g., Inktomi uses a cluster of hundreds of Sun Sparc workstations[5] with 75 GB of RAM and over 1 TB of spinning disk, and it crawls over 10 million pages a day.

In spite of these heroic efforts with high-end multiprocessors and exquisitely crafted crawling software, the largest crawls cover only 30-40% of the web, and refreshes take weeks to a month [2,22]. The overwhelming engineering challenges are in part due to the one-size-fits-all philosophy: Alta Vista and Inktomi try to cater to *every possible query* that might be made on the web. Although such services are invaluable for their broad coverage, the resulting diversity of content often snares all but the most craftily constructed queries in thousands of responses of little relevance or quality. Furthermore, the imminent explosion of web publication beyond North America and Europe, and beyond academic and corporate sites, will challenge even the most scalable solutions.

Compared to the web, development of the human brain has been tardy: it has grown ''only linearly'' from 400 to 1400 cubic centimeters in the last 3.5 million years. How do people avoid information overload? Serious web users adopt the strategy of filtering by *relevance* and *quality*. The growth of the web matters little to a physicist if at most a few dozen pages dealing with quantum electrodynamics are added or updated per week. Seasoned users also rarely roam aimlessly; they have bookmarked sites important to them, and their primary need is to expand and maintain a community around these examples while preserving the quality.

We argue that a giant, all-purpose crawl is neither necessary nor sufficient for this purpose. In our experience (§2), keyword queries cannot naturally locate resources relevant to specific topics. It is also unreasonable to have to first crawl and index 350 million pages in order to distill fifty good resources related to quantum electrodynamics! Much of this index would never be used, but, burdened by the responsibility of maintaining this huge index, the crawler would not be able to preferentially and frequently refresh and further explore relevant regions of the web. It might be argued that a central crawl amortizes work across multiple topics. But our results (§4) suggest that topical web exploration is efficient enough for distributed deployment.

Our contributions: In this paper, we describe a *Focused Crawler* which seeks, acquires, indexes, and maintains pages on a specific set of topics that represent a relatively narrow segment of the web. It 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. Thus, web content can be managed by a distributed team of focused crawlers, each specializing in one or a few topics. Each focused crawler will be far more nimble in detecting changes to pages within its focus than a crawler that is crawling the entire web. The focused crawler is guided by a classifier which learns to recognize relevance from examples embedded in a topic taxonomy, and a distiller which identifies topical vantage points on the web. We describe the architecture in §3 and our experiences in §4.

Eventually, our goal is to impose sufficient topical structure on the web so that powerful semi-structured query, analysis, and discovery are enabled. Here are some compelling examples:

Discovering linkage sociology:

Is there a hyperlink between the web page of a speed trap (traffic radar) maker and an auto insurance company? Apart from other bicycling pages, what topics are prominent in the neighborhood of bicycling pages? (*First aid* is one answer found by our system.)

Locating specialty sites:

Getting isolated pages, rather than comprehensive sites, is a common problem with web search. Now we can order *sites* according to the density of relevant pages found there. E.g., we can find the top five sites specializing in mountain biking.

Semi-supervised learning:

Human-supervised topic learning yields very high-quality filtering, but needs labor-intensive training. Finding specialty sites can quickly generate large amounts of additional training data with little effort.

Detecting community culture:

Simple statistics about the link graph reveal important information about the community of the focused topic, e.g., whether it is competitive or collaborative (§4), the typical time taken by a good resource to become popular, etc.

Estimating community timescales:

Simple queries can identify topical regions of the web that grow or change dramatically as against those that are relatively stable. This can be of great value to the web ontologists at Yahoo! or The Mining Company.

There is much awareness that for serious web users, focused *portholes* are more useful than generic *portals*[6]: ''The most interesting trend is the growing sense of natural limits, a recognition that covering a single galaxy can be more practical-and useful-than trying to cover the entire universe'' [16]. A focused crawler is an example-driven automatic porthole-generator. In a companion paper [8] we have proposed new HTTP infrastructure to support bidirectional hyperlinks to facilitate exploration of fine-grained communities. We feel that the ability to focus on a topical subgraph of the web, as in this paper, together with the ability to browse communities within that subgraph, will lead to significantly improved web resource discovery.

2 Focused crawler administration

Central to a focused crawler is a canonical topic taxonomy with examples. To run a specific instance, initial human input has to be provided in two forms. The user has to select and/or refine specific topic nodes in the taxonomy, and may also need to provide additional example URLs which serve as starting points for the crawl. In this section we give a user's view of the system.

2.1 Operation synopsis

Canonical taxonomy creation:

When the system is built, the classifier is pre-trained with a canonical taxonomy (such as Yahoo!, The Open Directory Project, The Virtual Library or The Mining Co.) and a corresponding set of examples. The canonical (coarse-grained) classification tree is part of the initial system.

Example collection:

The user collects URLs that are examples of her interest. These are submitted to the system, e.g., by importing her bookmarks file.

Taxonomy selection and refinement:

The system proposes the most common classes where the examples fit best. The user can choose and mark some of these classes as **good**. Sometimes, the user may find the taxonomy too coarse, and refine some categories and move documents from one category to another.

Interactive exploration:

The system also proposes additional URLs in a small neighborhood of the examples, that appear to be similar to the examples. (This can be regarded as a slow-speed, interactive startup phase.) The user may inspect and include some of these as examples. The steps thus far are illustrated in Figure 1(a).

Training:

The classifier integrates the refinements made by the user into its statistical class models.

Resource discovery:

At this stage the system is ready to perform resource discovery as described in the rest of the paper.

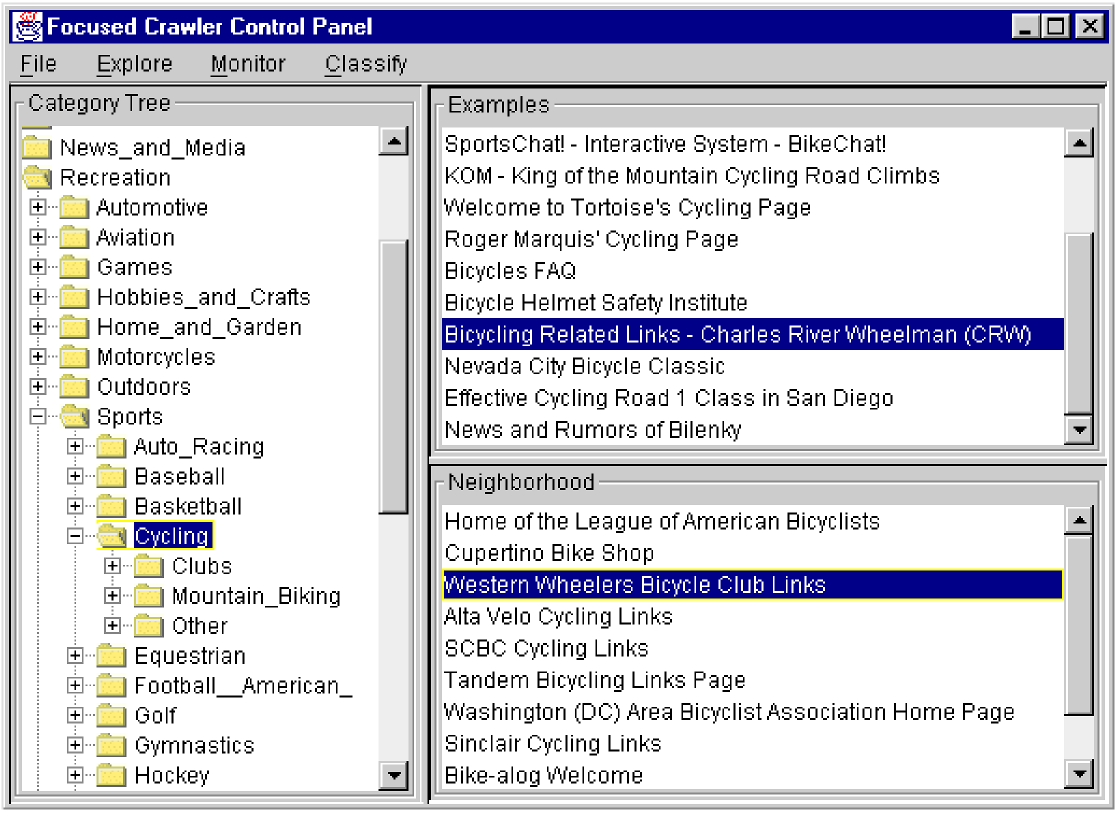
Distillation:

Intermittently and/or concurrently, the system runs a topic distillation algorithm to identify pages containing large numbers of relevant resource links, called *hubs*. The (re)visit priorities of these pages and immediate neighbors are raised.

Feedback:

Typically, the user inspects the system regularly. The system reports the most popular sites and resource lists, and the user can give feedback by marking them as useful or not. This feedback goes back to the classifier and distiller.

(a)


Focused Crawler Control Panel
File
Explore
Monitor
Classify
Category Tree
News_and_Media
Recreation
Automotive
Aviation
Games
Hobbies_and_Crafts
Home_and_Garden
Motorcycles
Outdoors
Sports
Auto_Racing
Baseball
Basketball
Cycling
Clubs
Mountain_Biking
Other
Equestrian
Football__American_
Golf
Gymnastics
Hockey
Examples
SportsChat! - Interactive System - BikeChat!
KOM - King of the Mountain Cycling Road Climbs
Welcome to Tortoise's Cycling Page
Roger Marquis' Cycling Page
Bicycles FAQ
Bicycle Helmet Safety Institute
Bicycling Related Links - Charles River Wheelman (CRW)
Nevada City Bicycle Classic
Effective Cycling Road 1 Class in San Diego
News and Rumors of Bilenky
Neighborhood
Home of the League of American Bicyclists
Cupertino Bike Shop
Western Wheelers Bicycle Club Links
Alta Velo Cycling Links
SCBC Cycling Links
Tandem Bicycling Links Page
Washington (DC) Area Bicyclist Association Home Page
Sinclair Cycling Links
Bike-alog Welcome

(b)

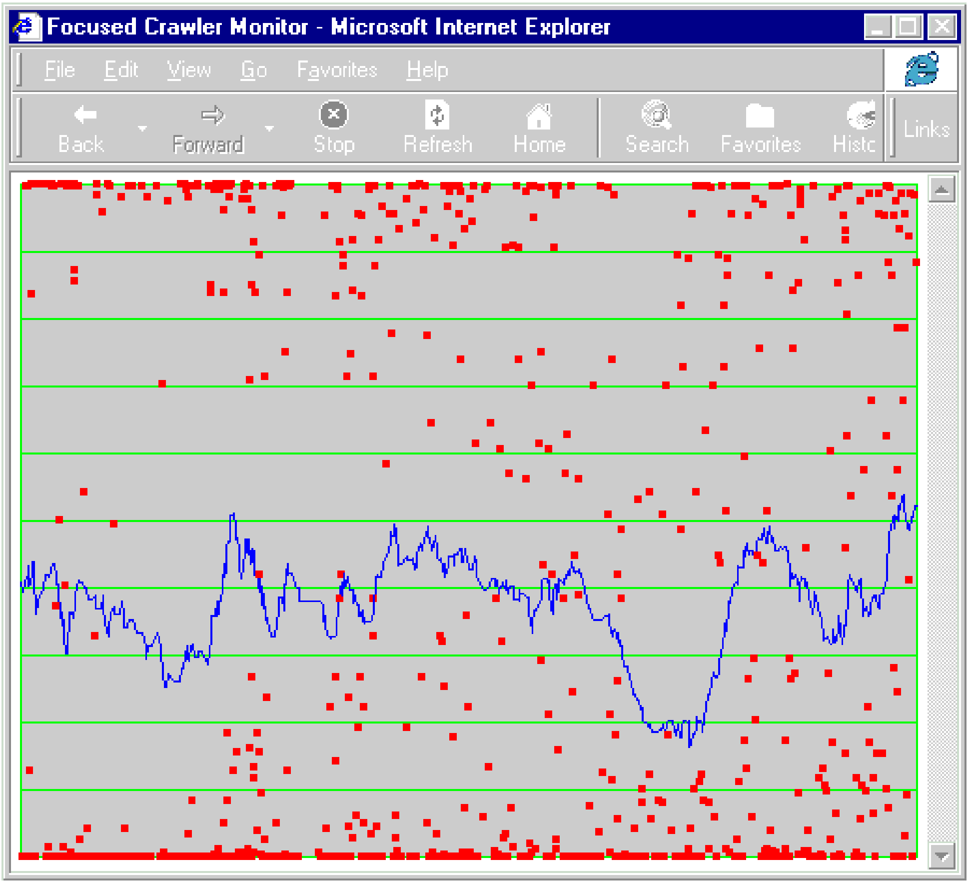

Focused Crawler Monitor - Microsoft Internet Explorer
File
Edit
View
Go
Favorites
Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
Histo
Links

Figure 1: Focused crawler administration and monitoring. (a) A sample session for configuring a crawl for ‘‘recreational bicycling’’ resources. (b) Applet for monitoring the recent relevant page acquisition rate of the focused crawler.

The user collects examples by browsing. The applet shown in Figure 1(a) monitors the page being rendered by the browser. Using the *Classify* menu, the user can make the classifier route the current page to the few best matching nodes in the category tree, marking all nodes on the way. After sufficient browsing, all marked nodes are presented to the user as candidates for focused crawling. The user selects some nodes and selects them, thereby marking them `good`. These are shown highlighted in the tree view. E.g., for the topic of ‘‘recreational bicycling,’’ two subtrees were found to be good choices. One (`/Recreation/Sports/Cycling`) is shown. The other was `/Business/Companies/Sports/Cycling`. Example sites that the master category system already knows about are shown in the upper right panel and can be viewed through a browser by clicking. When such a page is visited, the applet shows URLs of pages in the neighborhood of the example whose titles have many words in common with the most distinctive words of the topic [5,8]. Any pages thus found useful can also be added to the examples by dragging and dropping.

Sometimes the user may feel that the leaf nodes to which her examples have been assigned are still too broad and need to be refined. The tree view interface lets her create and move directories and populate them with examples. If major changes are made to the master category tree, some time is needed for the classifier to integrate the new structure into its models [5]. For our testbed with about 260,000 documents from Yahoo!, this takes a few hours. Smaller changes, such as moving of documents while keeping the tree unchanged, are interactive.

At this stage, the focused crawler can be started. It is a complex system which not only crawls tens of thousands of pages per hour, but makes decisions based on millions of arithmetic calculations per page. It is thus quite helpful for diagnostic purposes to visualize the status of the crawl graphically. We have developed an applet that maintains a plot of page relevance against time. In Figure 1(b), each red dot is a web page, which may be viewed in a browser window by clicking on the dot. The x-axis represents time. The y-axis is the relevance score (a probability value) between zero and one. The blue line is a smoothed moving average over a recent window of pages fetched. Continual refreshing introduces new points at the right edge, and the display scrolls the leftmost points off the screen.

If the page acquisition rate suddenly lowers, the right-to-left scrolling slows down. This can be made to raise an alarm (not implemented). Alternatively, the crawler may be getting many pages, but their relevance will be very low. The blue line will go down without significant recovery. This too can be made to raise an explicit alarm if necessary.

2.2 Justification and discussion

A different design is conceivable in which keyword search is used to locate an initial set of pages (using a giant crawl and index), expand this graph to a limited radius and then look for popular sites in the expanded graph using weighted degree measures [25,31,4,21,,3]. This approach was tried as a semi-automatic means to build a taxonomy like Yahoo!. For 966 topics picked from Yahoo!, keyword queries were constructed manually. E.g., the query for the topic `/Business/Companies/Electronics/PowerSupply` was `+"power suppl*" ßwitch* mode" smps -multiprocessor* üninterrupt* power suppl*" ups -parcel`. Typically, several query refinements were needed to match the quality of Yahoo! in blind user tests. The resulting queries were complex (as above) compared to the average Alta Vista query [29]. The above experiment used an average of 7.03 query terms and 4.34 operators (`+-*"`); an average Alta Vista query has only 2.35 terms and 0.41 operators. Query construction is not a one-time investment, because as pages on the topic are discovered, their additional vocabulary must be folded in manually into the query for continued discovery.

Yet another design is possible in which the focused crawler only uses the examples found by the user, but does not use a taxonomy with the pre-packaged examples. E.g., we can set up a simple two-class learning problem (relevant/irrelevant) for each focus topic. However, we have a few reasons to believe that our approach is more promising.

Better modeling of the negative class: In the two-class learning of text, characterization of the the negative class (e.g. ‘‘a page not about mutual funds’’) is often problematic. Using a taxonomy as we do, deals with this problem by describing the negative class as a union of positive classes. This is not merely a mental artifice, but it also affects the accuracy of learning algorithms significantly [26], because commonly used statistical models have large estimation errors on the diverse negative class.

Reuse of classifier training effort: Learning to recognize text documents is made difficult by the large dimensionality and consequent sparsity. It may be burdensome for every user to prepare enough sample documents to have an adequate number of positive and negative examples for learning her interest profile. The work of mapping the user’s interest onto a predefined set of categories, refining them when needed, will usually be significantly less than finding an adequate number of positive and negative examples. One may even envisage that a standards organization will design many ‘‘backbone taxonomies’’ for smaller groups to fill in and refine. A similar procedure is espoused for maintaining many web directories.

Discovery of related classes: The framework of a master taxonomy helps the user detect additional regions of the web that are topically related to her interest which were not naturally connected with her start set. As we shall see later, the focused crawler is quick to suggest that crawling only on *mutual funds* while forbidding *investment* in general will not work well, because the neighborhoods of these two topics comingle. It is able to do this because it can classify the web pages it encounters into other categories from the taxonomy, which would not be possible if the binary-classification approach were used.

Why is this significant? In addition to teaching the user about the relationship of her interests to other topics on the web, this capability is important for diagnostic purposes. In the *mutual funds* example, it is better to broaden the set of categories to those that provide a minimal covering of the interest topics, because doing so provides a higher degree of linkage, which means many more available paths for finding relevant pages. In such a scenario the crawling-priority relevance score and the final (for displaying to the user) relevance score will be determined differently. A natural way to expand the topic set for this purpose is to add some of the *parents* and/or *siblings* of the relevant topics from the taxonomy-another advantage over binary classification.

3 System architecture

The focused crawler has three main components: a **classifier** which makes relevance judgments on pages crawled to decide on link expansion, a **distiller** which determines a measure of centrality of crawled pages to determine visit priorities, and a **crawler** with dynamically reconfigurable priority controls which is governed by the classifier and distiller. A block diagram is shown in Figure 2. Here we briefly outline the basic processes. In subsequent work we have redesigned the modules on top of a relational database and efficiently integrated them [9].

Based on the discussion so far, we can summarize the role of the focused crawler in the following terms. We are given a directed hypertext graph G whose nodes are physically distributed. In this paper, G is the web. There is a cost for visiting any vertex (web page) of G. There is also a tree-shaped hierarchical topic directory C such as Yahoo!. Each topic node c Î C refers to some pages in G as examples. We denote the examples associated with topic c as $D(c)$. These pages can be preprocessed as desired by the system. The user's interest is characterized by a subset of topics C^* Ì C that is marked `good`. No good topic is an ancestor of another good topic. Ancestors of good topics are called `path` topics. Given a web page q, a measure of relevance $R_{C^*}(q)$ of q w.r.t. C^*, together with a method for computing it, must be specified to the system. C^* will be omitted if clear from the context. In this paper, we will use a probability measure 0 £ $R(q)$ £ 1. By definition, $R_{\{\mathrm{root}\}}(q) = 1$ "q. If $\{c_i\}$ are children of c_0, then å$_{c_i}$ $R_{c_i}(q) = R_{c_0}(q)$. The system starts by visiting all pages in $D(C^*)$. In each step, the system can inspect its current set V of visited pages and then choose to visit an unvisited page from the crawl frontier, corresponding to a hyperlink on one or more visited pages. Informally, the goal is to visit as many relevant pages and as few irrelevant pages as possible, i.e., to maximize average relevance. Therefore we seek to find V Ê $D(C^*)$ where V is reachable from $D(C^*)$ such that å$_V R(v)/|V|$ is maximized.

Our formulation would pose a hopeless problem if pages of all topics were finely dispersed all over the web. However, this is not likely to be the case. Citations signify deliberate judgment by the page author. Although some fraction of citations are noisy[7], most citations are to semantically related material. Thus the relevance of a page is a reasonable indicator of the relevance of its neighbors, although the reliability of this rule falls off rapidly with increasing radius on an average. This explains our use of the classifier. Secondly, multiple citations from a single document are likely to cite semantically related documents as well. This is why the distiller is used to identify pages with large numbers of links to relevant pages.

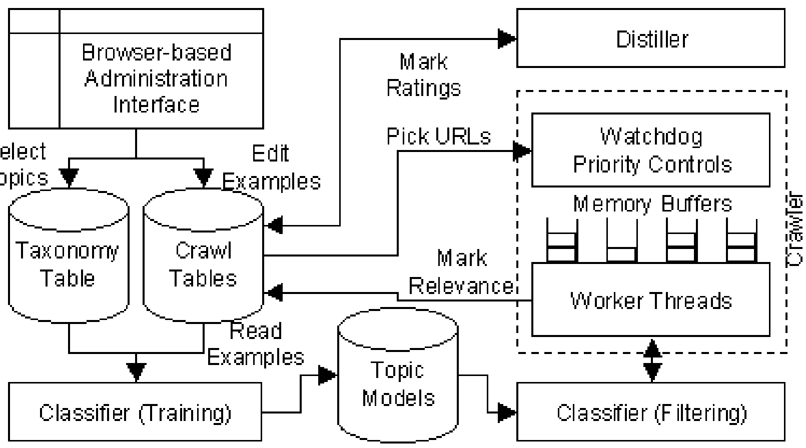


Figure 2: Block diagram of the focused crawler showing how the crawler, classifier and distiller are integrated.

3.1 Classification

Relevance is enforced on the focused crawler using a hypertext classifier [5,7]. We assume that the category taxonomy induces a hierarchical partition on web documents. (In real life, documents are often judged to belong to multiple categories. We plan to extend our model in future work.) Categories in the taxonomy tree, also called nodes, are denoted c. The predicate $\mathtt{good}(c)$ denotes whether a node c has been marked as good. By definition, for any document d, the probability that it was generated from the root category is 1. In general $\Pr[c|d] = \Pr[\mathtt{parent}(c)|d]\ \Pr[c|d,\mathtt{parent}(c)]$; this can be recursed using chain rule. Using Bayes rule we can write

$$\Pr[c|d,\mathtt{parent}(c)] = \Pr[c|\mathtt{parent}(c)]\ \Pr[d|c] \Big/ \sum_{c':\mathtt{parent}(c') = \mathtt{parent}(c)} \Pr[d|c'], \qquad ($$

where the sum ranges over all siblings c' of c.

To find $\Pr[d|c]$ we need a model for document generation. $\Pr[c|\mathtt{parent}(c)]$ define the *prior* distribution of documents. In our generation model, the page generator first decides the topic on which to write the document d by using these probabilities to pick a leaf node c^*. Each class, in particular c^*, has a die with as many faces as the number of unique words (terms, tokens) in the universe. Face t turns up with probability $\theta(c^*,t)$. The generator picks an arbitrary length $n(d)$ for the document. Then it repeatedly flips the die for c^*, and writes out the term corresponding to the face that turns up. A document is thus seen as a bag of words, without order information or inter-term correlation. If term t occurs $n(d,t)$ times, then

$$\Pr[d|c] = \binom{n(d)}{\{n(d,t)\}} \prod_{t \in d} \theta(c,t)^{n(d,t)}.$$

In spite of its simplicity, this model has been very successful. During crawling, the task is the reverse of generation: given a document, we seek to find the best leaf class c^*.

Two modes of focusing are possible with the classifier.

Hard focus rule:

While fetching a document d, the above formulation is used to find the leaf node c^* with the highest probability. If some ancestor of c^* has been marked $\mathtt{good}$ we allow future visitation of URLs found on d, otherwise the crawl is pruned at d.

Soft focus rule:

The probability that a page is relevant to the focused crawl is $R(d) = \sum_{\mathtt{good}(c)} \Pr[c|d]$, because a good node is never the ancestor of another. We do not eliminate any page a priori, but guess that the priority of visiting each neighbor of the current page d is the relevance of d. In case of multiple paths leading to a page we take the maximum of their relevance. When a neighbor is actually visited its score is updated.

3.2 Distillation

Relevance is not the only attribute used to evaluate a page while crawling. A long essay very relevant to the topic but without links is only a finishing point in the crawl. A good strategy for the crawler is to identify *hubs*: pages that are almost exclusively a collection of links to authoritative resources that are relevant to the topic.

Social network analysis [31] is concerned with the properties of graphs formed between entities such as people, organizations, papers, etc., through coauthoring, citations, mentoring, paying, telephoning, infecting, etc. *Prestige* is an important attribute of nodes in a social network, especially in the context of academic papers and web documents. The number of citations to paper u is a reasonable but crude measure of the prestige $p(u)$. A better measure is weighted citations, or the total prestige of papers that cite a paper. This notion is circular but can be resolved by an iterative eigen computation to find the fixpoint of $\mathbf{p} = \mathbf{E}\mathbf{p}$, where $\mathbf{E}$ is the directed adjacency matrix, as described by Katz [20] in 1953 and adapted to the web by Page et al [4].

Mizruchi et al [25] recognized that *centrality* in a social network can be disaggregated into *derived* and *reflected* centrality. They found two types of nodes: *bridges* which have high derived centrality, and *hubs* which link with good authorities and thereby have high reflected centrality. Kleinberg later exploited the same phenomenon on the Web to find hubs and authorities (bridges) [21].

Each node v has two corresponding scores, $h(v)$ and $a(v)$. Then the following iterations are repeated on the edge set E a suitable number of times: $a(v) \leftarrow \sum_{(u,v) \in E} h(u)$ and $h(u) \leftarrow \sum_{(u,v) \in E} a(v)$, interspersed with scaling the vectors $\mathbf{h}$ and $\mathbf{a}$ to unit length. This iteration embodies the circular definition that important hubs point to important authorities and vice versa.

For focused crawling, two important enhancements are needed: the edge weights must be carefully controlled and there should be a certain asymmetry in how we treat hubs and authorities. To appreciate the model that we will propose, observe that pages relevant to our interest refer to irrelevant pages and vice versa with appreciable frequency, owing to the diversity of web authorship. Pages of all topics point to Netscape and Free Speech Online. Conversely, many hubs are multi-topic in nature, e.g., a published bookmark file pointing to sports car sites as well as photography sites.

We will not only have non-unit edge weight but differentiate the forward and backward edge weights into two different matrices $\mathbf{E}_F$ and $\mathbf{E}_B$. We propose that the weight $\mathbf{E}_F[u,v]$ of edge (u,v) be the probability that u linked to v because v was relevant to the topic, i.e., $R(v)$. This has the effect of preventing leakage of prestige from relevant hubs to irrelevant authorities. Similarly, we propose that $\mathbf{E}_B[u,v]$ be set to $R(u)$, to prevent a relevant authority from reflecting prestige to an irrelevant hub. Finally, we will use a relevance threshold r to include potential authorities into the graph, although for hubs we have no such requirement. We include between 10 and 20% of the most relevant nodes; our results were not sensitive to the precise choice in this range. The remaining steps follow:

1. Construct the edge set E using only those links that are between pages on different sites, with forward and backward edge weights as above.
2. Perform the iterations using the weighted edges. Always restrict the authority set using the relevance threshold.

3.3 Integration with the crawler

The crawler has one *watchdog* thread and many worker threads. The watchdog is in charge of checking out new work from the crawl frontier, which is stored on disk. New work is passed to workers using shared memory buffers. Workers save details of newly explored pages in private per-worker disk structures. In bulk-synchronous fashion, workers are stopped, and their results are collected and integrated into the central pool of work.

The classifier is invoked by each thread as it encounters a new page. The *R* value computed is part of the page result mentioned above. The central work pool is a priority queue implemented using the Berkeley DB B-tree storage manager (http://www.sleepycat.com). For soft crawling, candidate URLs are ordered using a lexicographic combination

```
(numtries ascending, R descending),
```

where `numtries` is the number of times the crawler has already tried to fetch the page, with or without success. For hard crawling, the URLs that survive are picked in increasing order of `numtries`; for the same value of `numtries` the remaining order is arbitrary.

The crawler also populates the link graph kept on disk. Currently this consists of a forward and backward edge list, stored using the hash access method of Berkeley DB. Periodically, the crawler is stopped and the distiller is executed. This generates a number of top hubs to revisit. We also prepare to visit unvisited pages cited by the top hubs.

In ongoing work [9] we have reimplemented the system using a relational database to store the crawl frontier and facilitate dynamically changing prioritization strategies, and concurrent activity between the crawler, distiller and classifier. The integration also facilitates crawl monitoring and diagnostics using ad-hoc SQL queries.

4 Evaluation

In this section we will present our experiences with focused crawling. 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 provide anecdotes 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 topic. Whereas consensus has been forced in traditional IR benchmarks, such agreement would be very hard to arrive at in a reasonable manner in the case of the web. We will provide indirect evidence of robust coverage.

4.1 Experimental setup

The focused crawler is a C++ application running on a dual-processor 333 MHz Pentium-II PC with 256 MB of RAM and SCSI disk. Our test machines are connected through a half-duplex 10 MB/s Ethernet through the router to a SOCKS firewall machine. The firewall is connected to the ISP using full-duplex 10 MB/s SMDS over DS3. The ISP connects us to a 622 MB/s OC12 ATM backbone (UUNET High Performance Network).

A full-scale crawler never operates through a firewall. Although we had access to machines outside the firewall, we decided to demonstrate the viability of focused crawling by running it inside the firewall and consuming negligible network resources. We ran the crawler with relatively few threads compared to what it can handle to avoid disrupting firewall performance for others. Each instance of the crawler collected about 6000 URLs per hour.

We picked about twenty topics that could be represented by one or few nodes in a master category list derived from Yahoo!, such as gardening, mutual funds, cycling, HIV/AIDS, etc. Note that these are just category names and not queries; each category was trained with up to a few dozen starting example web pages. The main performance indicators were comparable for these and several other crawls. For concreteness we will present selected results from the above set. Most crawls were run for at least four hours. Some were left running for several days, mainly for stress-testing.

The crawls showed no signs of stagnation for lack of relevant pages, except for mutual funds. In that case, analyzing the crawl quickly indicated that many pages in the neighborhood of mutual funds were from `parent`(mutual funds), which was ‘‘investment’’ in general. These topics are so intimately mixed that an attempted crawl on one while rejecting the other was hopeless. Detecting and adapting to such scenarios automatically is an interesting area of future research.

In the following sections we make the following measurements.

- For a variety of topics, we study the absolute acquisition rate to see if it is high enough to warrant a focused crawl. We compare the distribution of relevance scores of soft focused, hard focused, and unfocused crawls.
- To judge the robustness of our system, we sampled disjoint fractions of the available set of ‘‘seed’’ URL’s and started separate crawls. We compare the rate of acquisition of relevant pages between the two crawlers. This is an indirect indicator of coverage.
- As another test of robustness, we ran the quality rating program on the crawls that started from the samples, and then measured the extent of overlap between the top rated pages and servers (IP addresses) found by the two crawlers.
- We present some of the top-rated URLs as anecdotal evidence of the quality of resource discovery. We show examples where promoting unvisited neighbors of top-rated hubs led to further acquisition of relevant pages.

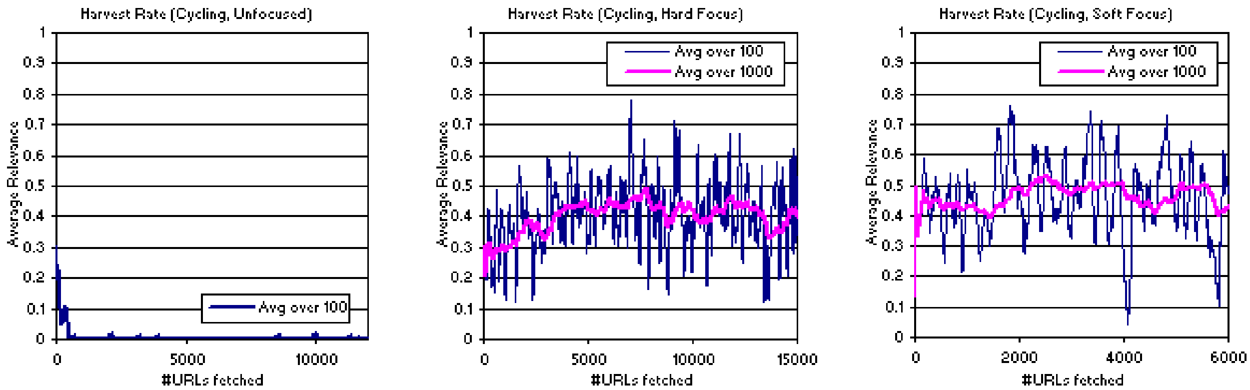


Figure 3: Rate of relevant page acquisition with a standard unfocused crawl, a hard focused crawl, and a soft focused crawl on the topic of bicyling.

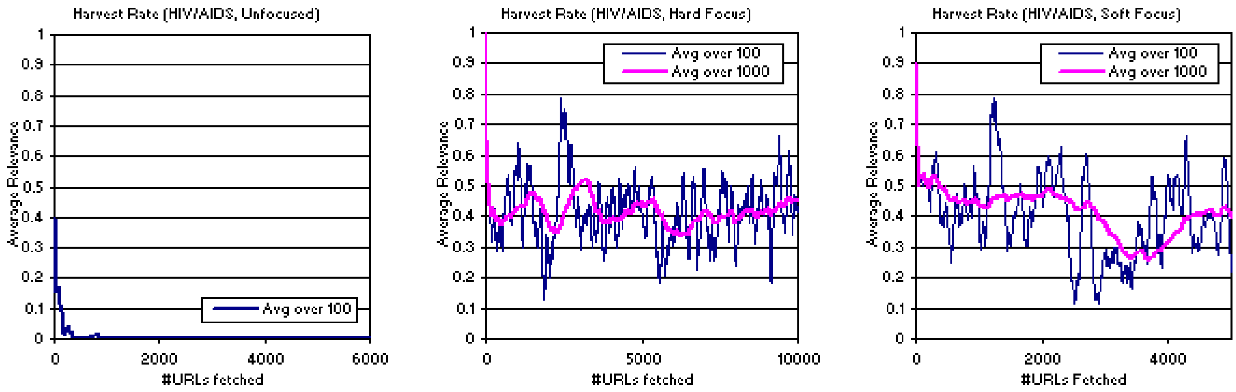


Figure 4: Rate of relevant page acquisition with a standard unfocused crawl, a hard focused crawl, and a soft focused crawl on the topic of HIV/AIDS.

4.2 Rate of harvesting relevant pages

Perhaps the most crucial evaluation of focused crawling is to measure the rate at which relevant pages are acquired, and how effectively irrelevant pages are filtered off from the crawl. This *harvest ratio* must be high, otherwise the focused crawler would spend a lot of time merely eliminating irrelevant pages, and it may be better to use an ordinary crawler instead!

It would be good to judge the relevance of the crawl by human inspection, even though it is subjective and inconsistent [23]. But this is not possible for the hundreds of thousands of pages our system crawled. Therefore we have to take recourse to running an automatic classifier over the collected pages. Specifically, we can use our classifier. It may appear that using the same classifier to guide the crawler and judge the relevance of crawled pages is flawed methodology, but it is not so. It is to be noted carefully that we are not, for instance, training and testing the classifier on the same set of documents, or checking the classifier's earlier work using the classifier itself. We are evaluating not the classifier but the basic crawling heuristic that neighbors of highly relevant pages tend to be relevant.

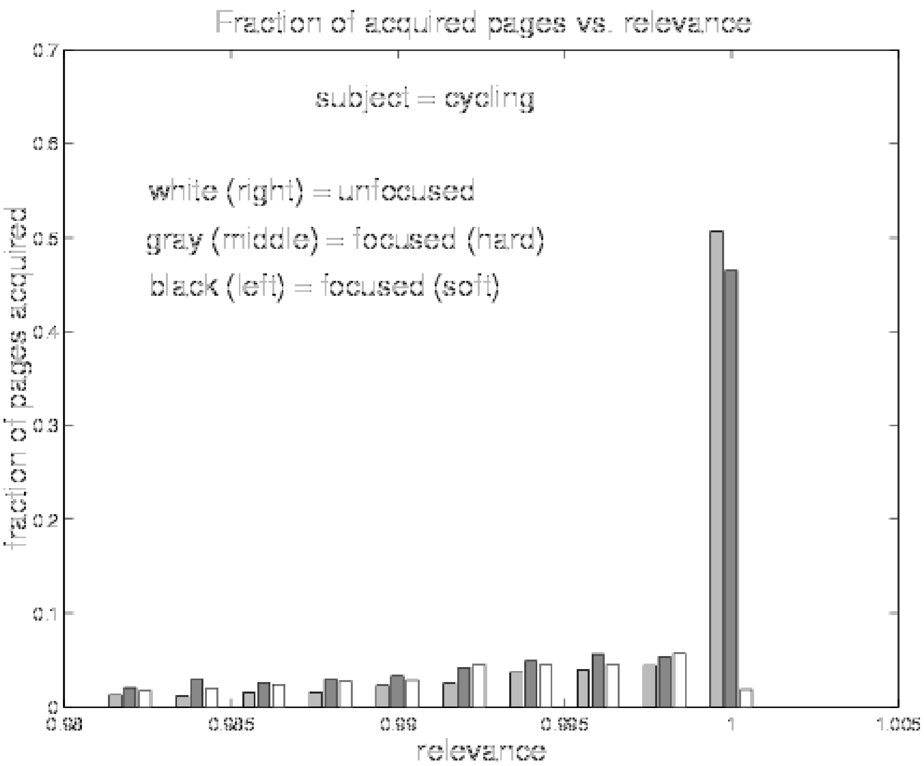
For each topic, three different crawls were done: *unfocused*, *soft focused* and *hard focused*. For each topic, the three crawls start out from the same set of a few dozen relevant URLs. These were collected by a keyword query at Alta Vista followed by traditional topic distillation and some screening by hand to eliminate irrelevant pages. In the unfocused case, the crawler fetches new URLs in pseudo-random order, and all out-links are registered for exploration. The pages are classified to find R, but no use is made of it except measurement. This will slow the crawl down a little. For this reason, and also because network load fluctuates greatly from experiment to experiment, in our results we present time not as wall-clock time, but as the number of URLs fetched so far.

The first column in Figures 3 and 4 shows the results of unfocused crawls for bicycling and HIV/AIDS. The x-axis shows the number of pages acquired (as a representative of real time). The y-axis shows a moving average of $R(u)$, where u represents pages collected within the window. It is immediately evident that focused crawling does not happen by accident; it has to be done very deliberately. The unfocused crawler starts out from the same set of dozens of highly relevant links as the focused crawler, but is completely lost within the next hundred page fetches: the relevance goes quickly to zero. This tends to happen even if we help it in various ways, such as disabling highly interconnected sites like Amazon.com.

In contrast, it is heartening to see in the second column of Figures 3 and 4 that the hard-focused crawls keep up a healthy pace of acquiring relevant pages over thousands of pages, in spite of some short-range rate fluctuations, which is expected. On an average, between a third and half of all page fetches result in success over the first several thousand fetches, and there seems to be no sign of stagnation. This rate was in fact higher than what we had hoped for. Similar observations hold for the soft focused crawler, shown in the third column.

Given that none of the crawls approached stagnation, it is difficult to compare between hard and soft focusing; they both do very well. For cycling, the hard crawler takes a little while to warm up because it loses some good opportunities to expand near-match pages. We believe the soft crawler is more robust, but needs more skill to monitor and guard against unwanted topic diffusion. The main technical problem in doing this is to distinguish between a noisy vs. systematic drop in relevance.

Figure 5 explain the earlier time-traces by showing the distribution of relevance of pages. Pages obtained by focused crawling show a very sharp peak at the highest possible relevance value, whereas the unfocused crawler shows essentially a flat distribution of relevance. It also appears (for cycling) that soft focusing can ‘‘tunnel through’’ mediocre pages to get slightly better pages than hard focusing.



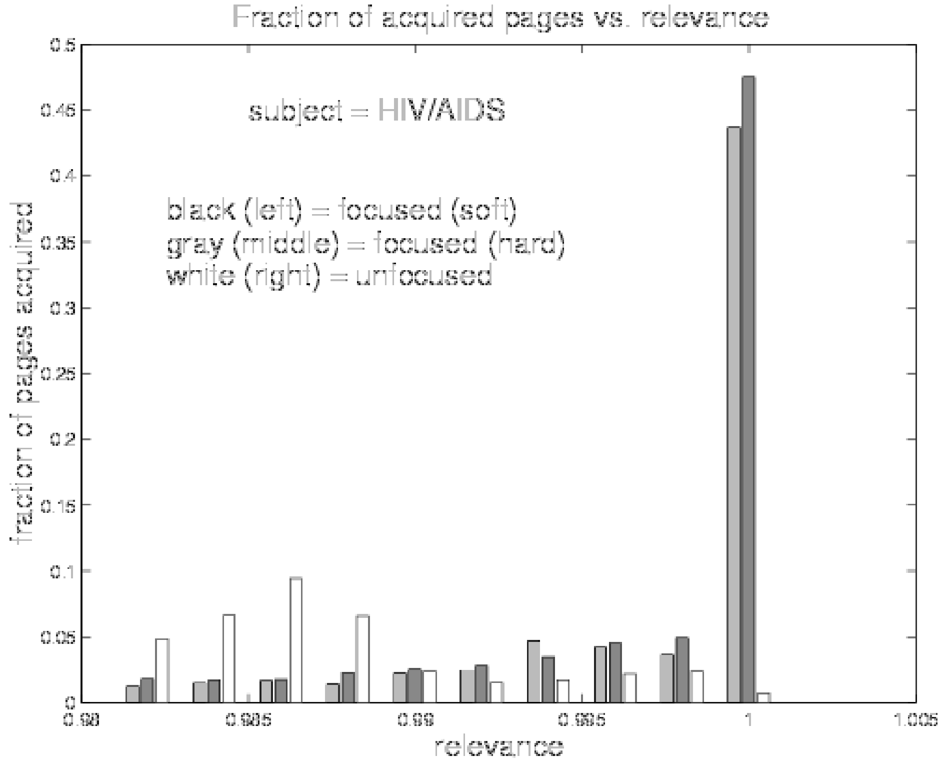


Figure 5: Distribution of relevance scores in the bicycling and HIV/AIDS crawls from the three crawlers.

4.3 Robustness of acquisition

Another important indicator of the robustness of a focused crawler is the ability to ramp up to and maintain a healthy acquisition rate without being too sensitive on the start set. To test this, we took the set of starting URL's and sampled subsets uniformly at random. We picked two disjoint random subsets each having about 30% of the starting URLs. For each subset, a different focused crawl was launched (at different times).

We will present two quantities. First we will measure the overlap of URLs crawled by the two crawlers. We will use bicycling and mutual funds as examples. The overlap is measured along time t, which is measured by counting the number of URLs fetched. (Direct comparison of wall-clock time is less meaningful owing to fluctuating network performance.) At any time t, the crawlers have collected URL sets $U_1(t)$ and $U_2(t)$. We plot $|U_1(t)\ \text{Ç}U_2(t)|/|U_1(t)|$ and $|U_1(t)\ \text{Ç}U_2(t)|/|U_2(t)|$ along time t (note that $|U_1(t)| = |U_2(t)|$ and therefore there is only one line in this case). Sample results are shown in Figure 6.

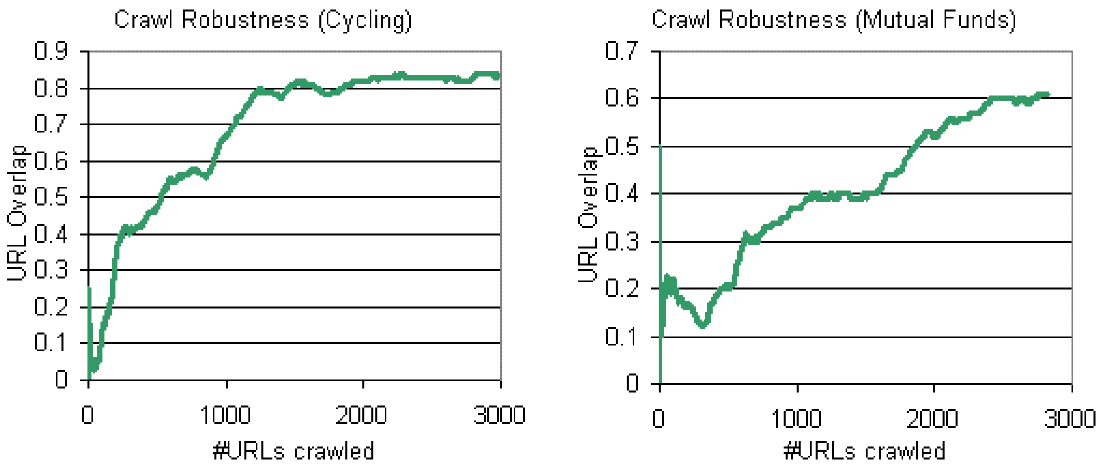


Figure 6: Overlap of URLs crawled by two soft focused crawlers starting from randomly sampled seed sets on bicycling and mutual funds.

We picked the two topics specifically because we wanted to study one co-operative community like bicycling and one competitive domain like investing and mutual funds. For cycling, the intersection between the set of URLs crawled grew rapidly to 90%. For mutual funds, it grew to over 60%. This confirmed our intuition about the two communities. The steady growth in overlap is heartening news, although it is a statement primarily about web behavior, not the focused crawler. It means that the choice of starting points is not critical for the success of focused crawling. We do have to double-check one thing, however. What if for reasons unknown, both crawlers started crawling pages out of one common site as soon as they reached there? This fear turns out to be ill-founded: a plot of the extent to which *IP-addresses* visited by the two cralers overlap against time shows generous visits to new IP-addresses as well as a healthy increase in the intersection of server IP-addresses. The intersections are plotted against time by first lining up the URL's fetched by each crawler side-by-side, then deriving the two sequences of IP-addresses visited, $S_1(t)$ and $S_2(t)$, and computing $|S_1(t)\ \text{Ç}S_2(t)|/|S_1(t)|$ and $|S_1(t)\ \text{Ç}S_2(t)|/|S_2(t)|$ for each t. In this case $|S_1(t)|$ is in general different from $|S_2(t)|$. The results are shown in Figures 7.

Results were similar with other topics. Whereas these results do not imply perfect coverage, they do indicate that core topical communities are fairly coherent, and emerge naturally from the crawl independent of the starting set. It would be interesting to stress the robustness by starting from smaller and smaller URL samples.

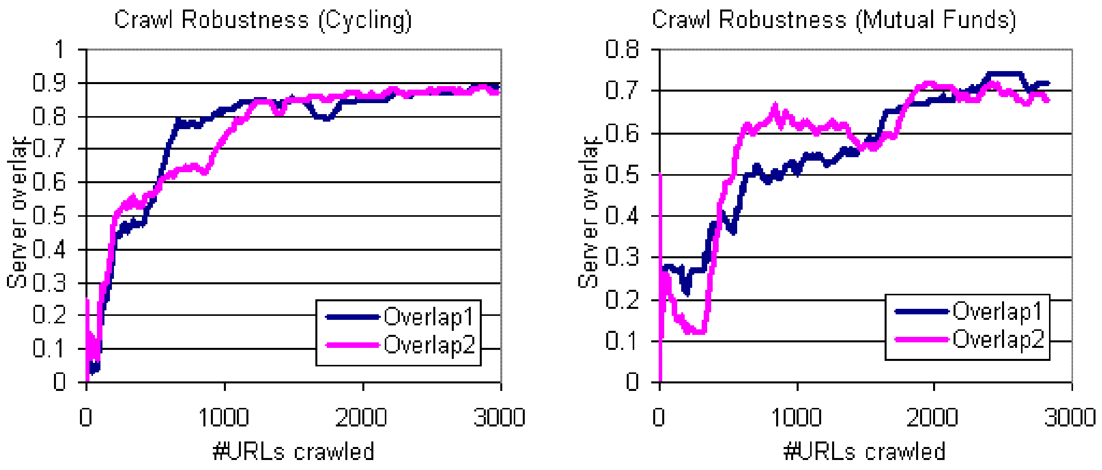


Figure 7: Overlap of servers crawled by two soft focused crawlers starting from randomly sampled seed sets on bicycling and mutual funds.

4.4 Robustness of resource discovery

Overlap in the set of servers and URLs crawled is a good indicator of inherent stability of the focused crawler. However, we wanted to also check that the topical subgraph of the web that is built by the focused crawler leads to robust estimations of popularity (estimated along the lines of recent topic distillation work). To do this we again used the two sets of crawlers that started from random samples of the available seed set. After acquiring 10,000 pages, we ran the popularity/quality rating algorithm with 50 iterations and produced a list of top ''authorities'' (as defined in HITS [21]). Then we measured the intersection of server IP-addresses in the top 25. We picked addresses rather than URL's because many pages, especially in mutual funds and HIV/AIDS are heavily frames enabled and have slight variants in URL. The results are shown in Figure 8. We see that in spite of slight local rank perturbations, the most popular sites are identified jointly by both runs of the focused crawler, although it started from different seed sets.

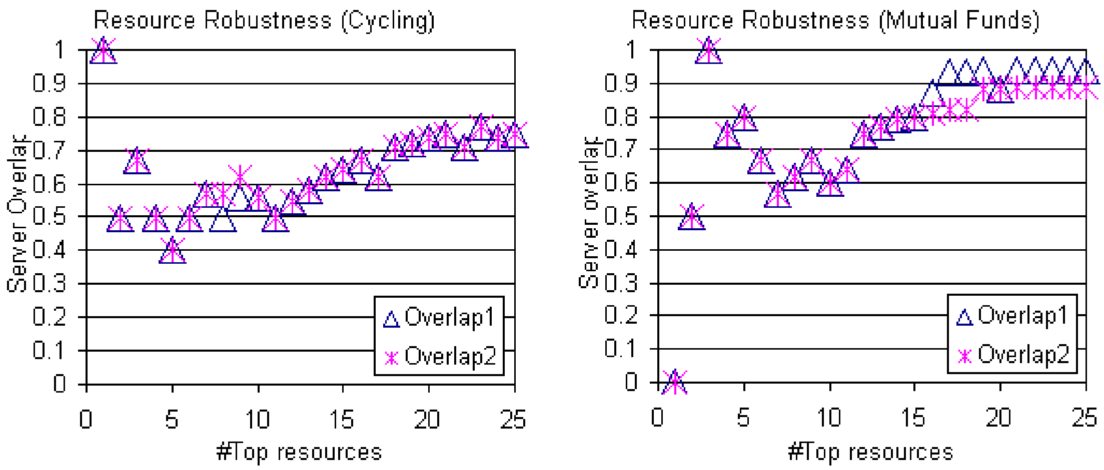


Figure 8: Overlap of the 100 best rated servers crawled by two soft focused crawlers starting from randomly sampled seed sets on cycling and mutual funds.

4.5 How remote are good resources?

Now we take a hard look at the following question: is the focused crawl doing any real exploration, or were the resources, specifically, the highly rated ones, within one or two links of the start set, or worse, *in* the start set? In Figure 9 we plot histograms of the number of servers in the 100 most popular ones that are a given radius away from the start set of URLs. We see a large number of servers at large distances from the start set, upto 10 links and beyond. Millions of pages are within 10 links from almost any page on the web. Thus the focused crawler is doing non-trivial work in pursuing only certain paths and pruning others. (There is some probability that the distances we report are pessimistic, and shorter paths to the best sites exist that were missed by the crawlers. Given we crawl best first on relevance, and that we tried multiple distinct seed sets, this should be rare.) Since our seed sets were collected using Alta Vista and HITS [21], this result also establishes the need to explore out aggressively from keyword-based and limited-radius search for resources.

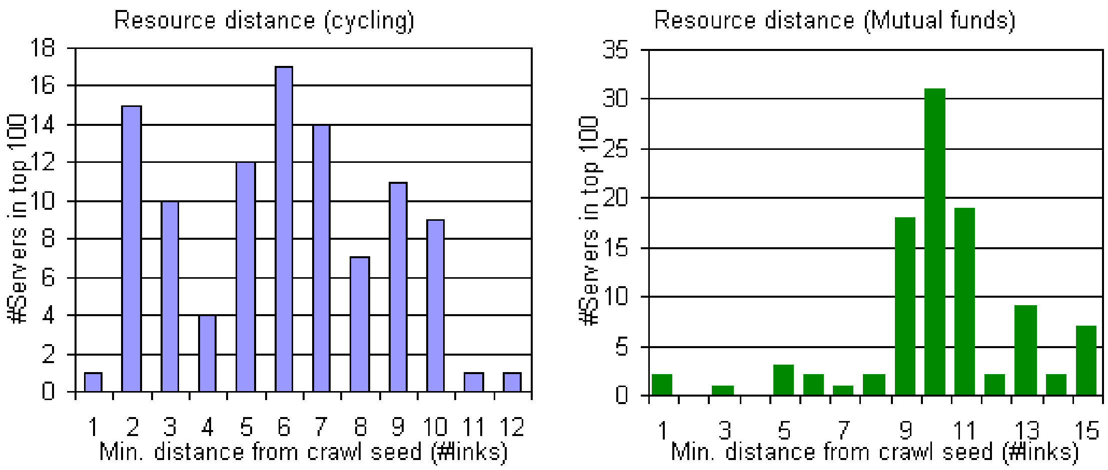


Figure 9: Distance in number of links from the seed set to the 100 most popular sites on cycling and mutual funds. The peak around 10 links for mutual funds is because great hubs were found around that distance.

A glance at the two histograms exposes the cooperative and competitive nature of the two communities. Cycling organizations are inclusive and social. Hence good hubs (consequently, authorities) are found at a variety of link distances. In contrast quite some exploration was needed for mutual funds and investment until good hubs were found, at radius 8-9. The focused crawler is good at pointing out these features.

4.6 Distillation anecdotes

Many post-processing operations on a focused crawl may be useful, such as clustering, indexing, etc. One more piece of evidence that a focused crawl is qualitatively better at resource discovery can be obtained by presenting the results of the distiller. Since we restrict the authority subgraph to only highly relevant nodes, our hubs tend to be topically very pure. Nothing short of human judgement is adequate for evaluating the rating algorithm; we strongly encourage the reader to visit the URL's found by our system and shown in Figure 10 (verified to be accessible as of March 1, 1999).

We only presented a small number of our top-rated pages; the list continues into thousands of pages. Spot checking failed to reveal irrelevant pages up to the first few hundred links. We were impressed that we could find over three thousand relevant sites within only an hour of focused crawling per topic using a desktop PC and starting with a few dozen URLs. The system did not need to consult any additional large keyword or link indices of the web, such as Alta Vista or Inktomi. Furthermore, almost half of our crawler's effort was useful from the point of view of the topics of interest.

Cycling	**HIV/AIDS**
http://www.truesport.com/Bike/links.htm	http://www.stopaids.org/Otherorgs.html
http://reality.sgi.com/employees/billh_hampton/jrvs/links.html	http://www-hsl.mcmaster.ca/tomflem/aic
http://www.acs.ucalgary.ca/~bentley/mark_links.html	http://www.ahandyguide.com/cat1/a/a66
http://www.cascade.org/links.html	http://www.iohk.com/UserPages/mlau/ai
http://www.bikeride.com/links/road_racing.asp	http://daphne.palomar.edu/library/subjec
http://www.htcomp.net/gladu/'drome/	http://www.us.unaids.org/highband/link.
http://www.tbra.org/links.shtml	http://www.ashastd.org/links/hivlinks.ht
http://www.islandnet.com/~ngs/SVCyclingLinks.html	http://www.hivresourcegroup.org/spd.ht
http://www.novit.no/dahls/Cycling/hotlist.html	http://allpaths.com/rainbow/aids.htm
http://members.aol.com/velodromes/MajorTaylor/links.htm	http://www.qrd.org/qrd/aids/
http://www.nashville.com/~mbc/mbc.html	http://www.teleport.com/~celinec/aids.sl
http://www.bikindex.com/bi/races.asp	http://www.aids.wustl.edu/aids/inet.html
http://www.louisvillebicycleclub.org/links.htm	http://virology.science.org/aids.html
http://world.std.com/~nebikclb/misc/netresources.html	http://www.metrokc.gov/health/apu/links
http://crisny.org/not-for-profit/cbrc/links.htm	http://www.sfaf.org/links.html
http://members.aol.com/velodromes/index.htm	http://www.aaas.org/science/aidslink.htm

Figure 10: Example hubs found by our relevance-conscious topic-distillation after crawling 6000 URLS (about an hour). The reader is strongly encouraged to visit these URLs.

4.7 Effect of distillation on crawling

The purpose of distillation in a focused crawler is not only an end-goal, but also a further enhancement to the crawling process. It sometimes happens that a very relevant page is abandoned after mis-classification, for example, when the page has many image-maps and very little text, and/or the statistical classifier makes a mistake. After running the distiller, it is quite easy to look for unvisited citations from the top hubs. E.g., performing this step with the HIV/AIDS hubs gives us the following unvisited URLs (that the reader is encouraged to visit):

http://www.planetout.com	http://gpawww.who.ch/gpahome.htm

http://www.actupny.org	http://www.gaypoz.com
http://www.users.dircon.co.uk/ eking/index.htm	http://aids.uspto.gov/AIDS/access/browse.html
http://www.aidsinfonyc.org	http://www.medibolics.com/nelson/index.htm

Many of these turned out to be relevant and worth crawling. We can now update the visit priority of these neglected neighbors to, say, the maximum possible value and restart the crawl. This process can be automated to run interspersed with normal crawling activity.

4.8 Summary

We have presented evidence in this section that focused crawling is capable of steadily collecting relevant resources and identifying popular, high-content sites from the crawl, as well as regions of high relevance, to guide itself. It is robust to different starting conditions, and finds good resources that are quite far from its starting point. In comparison, standard crawlers get quickly lost in the noise, even when starting from the same URLs. We end this section with two obervations that come out of all these measurements:

- The web graph is *rapidly mixing* w.r.t. topics: random links lead to random topics within an extremely short radius.
- At the same time, there exist long paths and large subgraphs where topical coherence persists.

These observations are not necessarily contradictory, and this is exactly what makes focused crawling worth doing.

5 Related work

Traditionally, machine learning techniques have been used to design filtering agents. WebWatcher [18] and HotList and ColdList [27] are examples of such filtering programs. Ackerman et al describe similar techniques [1]. In contrast to our technique, new pages are acquired in some of these systems by first extracting features that discriminate the hotlist from the coldlist and then using these features for posing keyword queries to standard web search engines. In the context of query refinement, two-way interactive classifiers have been used for relevance feedback. None of these systems deal with filtering at the data acquisition level, and for a large taxonomy.

Early web crawlers simply followed every link acquiring more and more pages. Crawlers and agents have grown more sophisticated [11]. To our knowledge the earliest example of using a query to direct a web crawl is the *Fish Search* system [14]. Similar results are reported for the WebCrawler [11]), Shark Search [17], and by Chen et al [10]. The focused crawler is different in using a topic taxonomy, learning from example, and using graph distillation to track topical hubs.

Ordinary search engines and directories are called *portals* or entry points into the web. There is growing consensus that *portholes*-sites that specialize in specific topics-are often more useful than portals[8]. A few systems that gather specialized content have been very successful. Cho et al compare several crawl ordering schemes based on link degree, perceived prestige, and keyword matches on the Stanford University web [12]. Terveen and Hill use similar techniques to discover related ‘‘clans’’ of web pages [30]. Ahoy! (http://www.cs.washington.edu/research/ahoy) [15,28] is a homepage search service based on a crawler specially tuned to locate homepages. Cora (http://www.cora.jprc.com/) is a search engine for computer science research papers, based on a crawler trained to extract such papers from a given list of starting points at suitable department and universities. These are special cases of the general example- and topic-driven automatic web exploration that we undertake.

Social networks have been analyzed for decades to find nodes with high prestige and reflected prestige [20,25,31]. Similar to PageRank [4], HITS [21], CLEVER [6], topic distillation [3] and link-based similarity search [13], we use social network analysis as a subroutine in our system. However, there are several important distinctions. Our distiller integrates topical content into the link graph model. PageRank has no notion of page content, and HITS and CLEVER explore the web to a preset radius (typically, 1) from the keyword query response. All involve pre-crawling and indexing the web. The focused crawler has no apriori radius cut-off for exploration, because it can use the classifier and distiller to guide itself. Thus the selection of relevant, high-quality pages happens directly as a goal-directed data acquisition process, not as post-processing or response to a query.

6 Conclusion

Generic crawlers and search engines are like public libraries: they try to cater to everyone, and do not specialize in specific areas. Serious web users are increasingly feeling a need for highly specialized and filtered ‘‘university research libraries’’ where they can explore their interest in depth [16,22]. Unlike public libraries, web libraries have little excuse not to specialize, because it is just a matter of locating and linking to resources.

We have demonstrated that goal-directed crawling is a powerful means for topical resource discovery. 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. Our system selects work very carefully from the crawl frontier. A consequence of the resulting efficiency is that it is feasible to crawl to a greater depth than would otherwise be possible. This may result in the discovery of some high-quality information resources that might have otherwise been overlooked. As Marchiori [24] has noted, the quality of such resources may not always be strongly related to simple link-based popularity.

A number of questions arise from our research. The harvest rate at the root is by definition 100%, and we have been seeing harvest rates of 30-40%. How does this depend on the specificity of the topic? At what specificity can focused crawls be sustained? Another issue to research is the sociology of citations between topics. While crawling topics described in this paper, we found a lot of anecdotal evidence that bicycle pages not only refer a lot to other bicycle pages, but also refer significantly more than one might expect to red-cross and first-aid pages. Similarly, HIV/AIDS pages often don’t directly refer to other HIV/AIDS pages, but refer to hospital home pages, which are more general. Discovering these kinds of relationships will give interesting insights in the way the web evolves.

Acknowledgements: We thank Tom Mitchell, Dan Oblinger and Steve Gates for helpful discussions; Myron Flickner for generously contributing disks and computers; David Gibson for helping with the Java user interface; and Sunita Sarawagi, Amit Somani and Kiran Mehta for advice with disk data structures.

References

[1] M. Ackerman, D. Billsus, S. Gaffney, S. Hettich, G. Khoo, D. Kim, R. Klefstad, C. Lowe, A. Ludeman, J. Muramatsu, K. Omori, M. Pazzani, D. Semler, B. Starr, and P. Yap. Learning probabilistic user profiles: Applications to finding interesting web sites, notifying users of relevant changes to web pages, and locating grant opportunities. *AI Magazine*, 18(2):47-56, 1997. Online at http://www.ics.uci.edu/ pazzani/Publications/AI-MAG.pdf.

[2] K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. In *Proceedings of the 7th World-Wide Web Conference (WWW7)*, 1998. Online at http://www7.scu.edu.au/programme/fullpapers/1937/com1937.htm; also see an update at http://www.research.digital.com/SRC/whatsnew/sem.html.

[3] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR Conference on Research and Development in Information Retrieval*, volume 21. ACM, 1998. Online at ftp://ftp.digital.com/pub/DEC/SRC/publications/monika/sigir98.pdf.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW Conference*, volume 7, 1998. Online at http://google.stanford.edu/ backrub/google.html.

[5] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *VLDB Journal*, Aug. 1998. Invited paper.

[6] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th World-wide web conference (WWW7)*, 1998. Online at http://www7.scu.edu.au/programme/fullpapers/1898/com1898.html and at http://www.almaden.ibm.com/cs/people/pragh/www98/438.html.

[7] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD*. ACM, 1998. Online at http://www.cs.berkeley.edu/ soumen/sigmod98.ps.

[8] S. Chakrabarti, D. Gibson, and K. McCurley. Surfing the web backwards. In *8th World Wide Web Conference*, Toronto, Canada, May 1999.

[9] S. Chakrabarti, M. van den Berg, and B. Dom. Distributed hypertext resource discovery through examples. Submitted to VLDB, Feb. 1999.

[10] H. Chen, Y.-M. Chung, M. Ramsey, and C. C. Yang. A smart itsy bitsy spider for the web. *J. Am. Soc. Inf. Sci.*, 49(7):604-618, 1998.

[11] F.-C. Cheong. *Internet Agents: Spiders, Wanderers, Brokers and Bots*. New Riders Publishing, Indianapolis, Indiana, 1996. ISBN: 1-56205-463-5.

[12] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. In *7th World Wide Web Conference*, Brisbane, Australia, Apr. 1998. Online at http://www7.scu.edu.au/programme/fullpapers/1919/com1919.htm.

[13] J. Dean and M. R. Henzinger. Finding related pages in the world wide web. In *8th World Wide Web Conference*, Toronto, May 1999.

[14] P. DeBra and R. Post. Information retrieval in the world-wide web: Making client-based searching feasible. In *Proceedings of the First International World Wide Web Conference*, Geneva, Switzerland, 1994.

[15] O. Etzioni. Moving up the information food chain: Deploying softbots on the world wide web. In *Proceedings of AAAI-96*, 1996.

[16] D. Gillmor. Small portals prove that size matters. Tech column in San Jose Mercury News, Dec. 1998. Online at http://www.sjmercury.com/columnists/gillmor/docs/dg120698.htm and http://www.cs.berkeley.edu/ soumen/focus/DanGillmor19981206.htm.

[17] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalheim, and S. Ur. The Shark-Search algorithm-an application: Tailored web site mapping. In *7th World-Wide Web Conference*, Brisbane, Australia, Apr. 1998. Online at http://www7.scu.edu.au/programme/fullpapers/1849/com1849.htm.

[18] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A tour guide for the web. In *IJCAI*, Aug. 1997. Online at http://www.cs.cmu.edu/ webwatcher/ijcai97.ps.

[19] B. Kahle. Preserving the internet. *Scientific American*, Mar. 1997. Online at http://www.sciam.com/0397issue/0397kahle.html and http://www.alexa.com/ brewster/essays/sciam_article.html.

[20] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39-43, Mar. 1953.

[21] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998. Also appears as IBM Research Report RJ 10076(91892), and online at http://www.cs.cornell.edu/home/kleinber/auth.ps.

[22] S. Lawrence and C. L. Giles. Searching the world wide web. *Science*, 280:98-100, April 1998.

[23] S. Macskassy, A. Banerjee, B. Davidson, and H. Hirsh. Human performance on clustering web pages: A performance study. In *Knowledge Discovery and Data Mining*, volume 4, pages 264-268, 1998.

[24] M. Marchiori. The quest for correct information on the web: Hyper search engines. In *Sixth International World Wide Web Conference*, Santa Clara, Apr. 1997.

[25] M. S. Mizruchi, P. Mariolis, M. Schwartz, and B. Mintz. Techniques for disaggregating centrality scores in social networks. In N. B. Tuma, editor, *Sociological Methodology*, pages 26-48. Jossey-Bass, San Francisco, 1986.

[26] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 1999. Online at

http://www.cs.cmu.edu/ knigam/papers/emcat-mlj99.ps.gz.

[27] M. Pazzani, L. Nguyen, and S. Mantik. Learning from hotlists and coldlists: Towards a www information filtering and seeking agent. In *Seventh International Conference on Tools with Artificial Intelligence*, 1995. Online at http://www.ics.uci.edu/ pazzani/Publications/Coldlist.pdf.

[28] J. Shakes, M. Langheinrich, and O. Etzioni. Dynamic reference sifting: a case study in the homepage domain. In *Proceedings of the 6th World-Wide Web Conference (WWW6)*, 1997.

[29] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large AltaVista query log. Technical Report 1998-014, COMPAQ System Research Center, Oct. 1998. Online at http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1998-014.html.

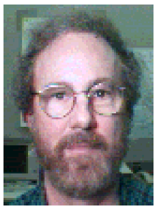
[30] L. Terveen and W. Hill. Finding and visualizing inter-site clan graphs. In *Computer Human Interaction (CHI)*, pages 448-455, Los Angeles, CA, Apr. 1998. ACM SIGCHI. Online at http://www.research.att.com/ terveen/chi98.htm and http://www.acm.org/pubs/articles/proceedings/chi/274644/p448-terveen/p448-terveen.pdf.

[31] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

Vitae

Soumen Chakrabarti received his B.Tech in Computer Science from the Indian Institute of Technology, Kharagpur, in 1991 and his M.S. and Ph.D. in Computer Science from the University of California, Berkeley in 1992 and 1996. He was a Research Staff Member at IBM Almaden Research Center between 1996 and 1999, and is now an Assistant Professor in the Department of Computer Science and Engineering at the Indian Institute of Technology, Bombay. His research interests include hypertext information retrieval, web analysis and data mining.

Martin van den Berg received a M.S. in Theoretical Physics in 1989 and a Ph.D. in Computational Linguistics in 1996 from the University of Amsterdam. His research interests are in the study of structural and semantic aspects of natural language discourse using formal logics, information retrieval and the study of dynamic logic. Currently he works for the FX Palo Alto Laboratory, where he does research on semi-automatic text understanding. Prior to that he spent a year as postdoctoral fellow at the IBM Almaden Research Center.



Byron Dom received a Ph.D. in Applied Physics from The Catholic University of America in 1977. He is currently manager of Information Management Principles at IBM Almaden Research Center, where he has been a Research Staff Member since 1987. Prior to that he was a Research Staff Member at the IBM T. J. Watson Research Center His research interests are in information retrieval, machine learning, computer vision, and information theory. He has led several successful projects to develop automatic inspection systems for aspects of computer manufacturing. He is the recipient of several IBM awards for technical excellence; has served as conference chair, session chair and on the program committees of several computer vision conferences; and has served as Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence.

Footnotes:

[1] Tested for viewing and printing with Netscape Communicator on Windows NT. The printed version is available in PDF format from http://www.cs.berkeley.edu/~soumen/doc/www99focus/

[2] Work partly done at IBM Almaden.

[3] Work done at IBM Almaden

[4] http://www.altavista.com/av/content/about_our_technology.htm

[5] http://www.inktomi.com/Tech/CoupClustWhitePap.html

[6] See excerpts from the press at http://www.cs.berkeley.edu/ soumen/focus/.

[7] ‘‘This page is best viewed using Netscape,’’ or ‘‘support Free Speech Online.’’

[8] See the press articles archived at http://www.cs.berkeley.edu/ soumen/focus/.

File translated from TEX by TTH, version 2.01.
On 29 Mar 1999, 18:11.