Evaluating Topic-Driven Web Crawlers

Filippo Menczer[@], Gautam Pant[@], Padmini Srinivasan[@!]
[@]Department of Management Sciences
[!]School of Library and Information Science
The University of Iowa
Iowa City, IA 52242

{filippo-menczer, gautam-pant, padmini-srinivasan}@uiowa.edu

Miguel E. Ruiz
TextWise
Dey Centennial Plaza
Syracuse, NY 13202

mruiz@textwise.com

ABSTRACT

Due to limited bandwidth, storage, and computational resources, and to the dynamic nature of the Web, search engines cannot index every Web page, and even the covered portion of the Web cannot be monitored continuously for changes. Therefore it is essential to develop effective crawling strategies to prioritize the pages to be indexed. The issue is even more important for topic-specific search engines, where crawlers must make additional decisions based on the relevance of visited pages. However, it is difficult to evaluate alternative crawling strategies because relevant sets are unknown and the search space is changing. We propose three different methods to evaluate crawling strategies. We apply the proposed metrics to compare three topic-driven crawling algorithms based on similarity ranking, link analysis, and adaptive agents.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*performance evaluation*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*graph and tree search strategies*

General Terms

Performance, Measurement, Algorithms, Design

Keywords

Best-first search, focused crawlers, InfoSpiders, PageRank, performance metrics, topic driven crawling, Web information retrieval

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for pro£t or commercial advantage and that copies bear this notice and the full citation on the £rst page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior speci£c permission and/or a fee.
SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA.
Copyright 2001 ACM 1-58113-331-6/01/0009 ...$5.00.

1. INTRODUCTION

Crawler programs designed to retrieve Web pages are essential for many of the key applications served by the Web. Starting with the early query based crawlers such as Fish Search [8] and WebCrawler [17], crawlers have today become the basis not only for the large search engines but also for the many specialized services such as investment portals, competitive intelligence tools, and scientific paper repositories.

Crawlers exploit the Web's hyperlinked structure to retrieve new pages by traversing links from previously retrieved ones. As pages are fetched, their outward links are added to a list of unvisited pages, which is referred to as the crawl frontier. A key challenge during the progress of a crawl is to identify the next most appropriate link to follow from its crawl frontier.

The algorithm to select the next link for traversal is necessarily tied to the goals of the crawler. A crawler that aims to index the Web as comprehensively as possible will make different kinds of decisions than one aiming to collect pages from university Web sites or one looking for pages about movie reviews. For the first, crawl order may not be important, the second may consider the structure of the URL to limit retrieved pages to the `.edu` subset, while the third may use similarity with the source page to guide link selection. Even crawlers serving the same goal may adopt different crawl strategies.

One research area that is gathering increasing momentum is the evaluation of crawlers. Our rich history of evaluation studies comparing retrieval algorithms in the non-Web context offers many evaluation methods and measures that may be applied towards this end. However, given that the dimensions of the crawler evaluation problem are dramatically different, the design of appropriate evaluation strategies is a valid challenge.

Our first goal is to explore three novel approaches for assessing and comparing topic driven crawlers. The three methods differ in how they assess the value of crawled pages: the first builds text classifiers, the second uses an independent retrieval system, in this case SMART, and the third applies a simple similarity metric to the dynamic set of crawled pages. Our second goal is to apply this evaluation framework to compare three types of crawlers. The first is Best-First search, which prioritizes links in the frontier based on the similarity between the query and the page where the link was found. The second crawler is based on PageRank,

the well known link-based algorithm used for ranking by the Google search engine [4]. The third crawler, InfoSpiders, is based on search agents that evaluate links via neural nets and adapt via an evolutionary algorithm [15]. One relatively unique aspect is that we explore assessment strategies within the constraints of limited resources.

2. EVALUATION OF CRAWLERS

In a general sense, a crawler may be evaluated on its ability to retrieve "good" pages. However, a major hurdle is the problem of recognizing these good pages. In an operational environment real users may judge the relevance of pages as these are crawled allowing us to determine if the crawl was successful or not. Unfortunately, meaningful experiments involving real users for assessing Web crawls are extremely problematic. For instance the very scale of the Web suggests that in order to obtain a reasonable notion of crawl effectiveness one must conduct a large number of crawls, i.e., involve a large number of users.

Secondly, crawls against the live Web pose serious time constraints. Therefore crawls other than short-lived ones will seem overly burdensome to the user. We may choose to avoid these time loads by showing the user the results of the full crawl — but this again limits the extent of the crawl. Next we may choose indirect methods such as inferring crawler strengths by assessing the applications that they support. However this assumes that the underlying crawlers are openly specified, and also prohibits the assessment of crawlers that are new.

Thus we argue that although obtaining user based evaluation results remains the ideal, at this juncture it is appropriate and important to seek user independent mechanisms to assess crawl performance. Moreover, in the not so distant future, the majority of the direct consumers of information is more likely to be Web agents working on behalf of humans and other Web agents than humans themselves. Thus it is quite reasonable to explore crawlers in a context where the parameters of crawl time and crawl distance may be beyond the limits of human acceptance imposed by user based experimentation.

Since we are not involving real users, we use topics instead of queries, each represented by a collection of seed URLs. It is clear that we are simplifying issues by moving from queries to topics. For example, we loose any clues to user context and goals that queries may provide. However, this approach of starting with seed URLs is increasingly common in crawler research. We assume that if a page is on topic then it is a "good" page. There are obvious limitations with this assumption. Topicality, although necessary, may not be a sufficient condition for user relevance. For example, a user who has already viewed a topical page may not consider it relevant since it lacks novelty. While we do not underrate these criteria, given the reasons stated above we choose to focus only on topicality as an indicator of relevance for the extent of this research.

3. RELATED WORK

Recent research reveals several innovative measures of performance. We first observe that there are two dimensions in the assessment process. First we need a measure of the crawled page's importance and second we need a method to summarize performance across a set of crawled pages.

3.1 Page importance

At a general level page importance measures are of two types: link-based and similarity-based. Note that for our purposes "similarity" refers to content-based similarity. Similarity based measures show a range of sophistication. Cho *et al.* [7] explore a rather simple similarity measure: the presence of a single word such as "computer" in the title or above a frequency threshold in the body of the page is enough to indicate a relevant page. Amento *et al.* [1] compute similarity between a page's vector and the centroid of the seed documents as one of their measures of page quality. Similarity as measured using page text [3] or the words surrounding a link [6] may also be used to augment what are primarily link based measures. Chakrabarti *et al.* take a unique approach, which is to apply classifiers built using positive and negative example pages to determine page importance [6]. We also explore classifiers for page evaluation, however there are differences as described below.

In-degree, out-degree, PageRank, hubs and authorities are the more commonly used link-based page importance measures [1, 2, 3, 6, 7]. For example, Cho *et al.* consider pages with PageRank above a specified threshold as being relevant to the query [7]. Kleinberg's recursive notion of hubs and authorities [13] has been extended by several others. For example, edge weights are considered important [6] and so are edges that connect different sites [1, 3, 6]. Link based quality metrics rely on the generally reasonable notion of links reflecting the credibility of the target pages. Interestingly Amento *et al.* shows that in-degree, authority and PageRank are effective at identifying high quality pages as judged by human experts [1].

Most of the link-based methods such as PageRank were designed to operate within a "neighborhood graph" of a query and thus implicitly recognize content-based criteria [13]. For instance in Fetuccino [2] and InfoSpiders [15] the crawl starting points are obtained via CLEVER (IBM's engine based on HITS) or any search engine, respectively. Various combinations of similarity and link-based criteria have also been suggested to evaluate links and guide crawlers, such as looking at the words surrounding a link [5, 15] and building belief networks [21]. One exception is the set of experiments performed by Cho *et al.*, wherein purely link based metrics were used to direct as well as assess some of their crawls [7].

3.2 Summary Analysis

Given a particular measure of page importance (PageRank, Backlink Count or similarity) Cho *et al.* [7] examine the percentage of important pages retrieved over the progress of the crawl. Menczer *et al.* [15] measure search length, the number of pages crawled until some predetermined fraction of important pages have been visited.

Chakrabarti *et al.* [6] perform a similar analysis with classifiers for page importance, using what they call the average "harvest ratio," i.e., the average number of relevant pages retrieved over different time slices of the crawl. The authors also measure crawler robustness using identical crawlers that are started on different subsets of the seed set. Robustness is then measured by the overlap in URLs as well as servers crawled over time. It appears that they treat robustness independently of page importance. Finally, they measure the "remoteness" of the authoritative pages crawled as their minimum distance (number of links) from the seed set. The

greater this distance, the greater the ability of the crawler to focus its search.

4. EVALUATION METHODS

4.1 Assessment via Classi£ers

Our first approach for evaluating topic-driven crawlers is to use classifiers. We build a classifier for each of 100 topics using a training set. The positive examples in the training set for a topic P consists of all pages corresponding to P's seed URLs. The negative examples are the pages of the seed URLs for the other 99 topics. The one exception is that if topic Q and P share any URLs in their seed sets, then Q's URLs are excluded from P's negative example set. These classifiers are used to assess the newly crawled pages. A positively classified page is viewed as a "good" page for the topic. In this sense this measure may be viewed as similar to precision where content-based relevance is decided by the classifier.

We use Widrow-Hoff (WH), Exponentiated Gradient (EG), and Rocchio classifiers [23, 12, 11] with feature selection using Correlation Coefficient [16] to select the best 50 features for each topic. The optimal threshold is set by maximizing the F1 score [22] on the training set. Due to limited space we refer the reader to [14, 25] for details on the classifiers.

It may be observed that although Chakrabarti *et al.* also use classifiers to assess pages [6], our evaluation strategy differs in several ways. First, their classifier is dependent upon a hierarchy of topics. More specifically, their calculations involving the conditional probability $P(concept|document)$ depends upon the hierarchical structure linking concepts. In contrast, although we exploit the Yahoo hierarchy for topics, we would just as easily be able to work with an ordinary (i.e., non hierarchical) set of topics. Second, for some crawls they utilize the same classifiers to both guide and assess the crawl strategy. In our work we use classifiers only for evaluation, keeping it distinct from the crawler algorithm.

4.2 Assessment via a Retrieval System

Our second approach is to use an independent retrieval system to rank the crawled pages against the topic. Crawlers are then assessed by looking at when the "good" pages were fetched. A good crawler will retrieve high ranking pages earlier than the lower ranked ones. Note that the temporal position of the fetched page is of significance only if the crawl is being conducted live for a user. Instead if the intent is only to add the results of a crawl to a cumulative index (such as with a search engine) then temporal position is less interesting. It may be observed that it is not possible for a crawler to retrieve a page before traversing its minimum distance from the seed set. But this constraint is consistent across all tested crawlers since they share the same seed set of URLs. Since the independent system ranks the pages after the crawl has been completed it may utilize term weighting strategies involving inverse document frequency computed over all fetched pages, whereas the same information is not available while the crawl is in progress.

Clearly any reasonable retrieval system may be used for this purpose. We choose to use the SMART system [20] given its reputation and its widespread use in IR research. While a crawl is in progress we store the sequence in which the pages (each given a unique `id`) are fetched. For a given topic the pages retrieved by all the crawlers are pooled together and indexed for SMART. We use *atc* weights for both the topic and page vector weights. The *at* components of the weight for term k and page p are computed as:

$$w_{kp}^{at} = \frac{1}{2}\left(1 + \frac{f_{kp}}{\max_{t\in T, d\in S} f_{td}}\right) \cdot \ln\left(\frac{|S|}{n_k}\right)$$

where f_{kp} is the frequency of term k in document p, S is the set of all crawled pages, T is the set of all terms in S, and n_k is the number of pages containing k. The c component implies cosine normalization of the weights.

All of the crawled pages (irrespective of which crawlers found them) are ranked by SMART using the topic (cf. Section 5) as the query. We then calculate the RankScore of any crawl as the average rank of the pages fetched in the crawl:

$$RankScore(S(t), S) = \frac{1}{|S(t)|}\sum_{p\in S(t)} rank(p, S)$$

where $S(t)$ is the set of pages crawled up to time t and $rank(p, S)$ is the rank attributed by SMART to p within S. Since we use rank, the lower RankScore, the better the crawler. The best (lowest) RankScore is obtained when the crawler retrieves the `MAX_PAGES` pages best ranked by SMART as early as possible given their distance from the seed set (`MAX_PAGES` is the maximum number of pages that may be fetched by a crawler). In general, the RankScore measure is akin to search length in that we can determine the number of pages to fetch given that we wish to achieve a certain level of quality as measured via SMART.

4.3 Assessment via Mean Topic Similarity

Our third approach is related to the previous one, but simpler. We argue that a good crawler should remain in the vicinity of the topic *in vector space,* therefore we measure the average cosine similarity between the *tf*idf* vector of the topic and the *tf*idf* vector of each page visited up to a certain point in the crawl:

$$sim(q, S(t)) = \frac{1}{|S(t)|}\sum_{p\in S(t)} \frac{\sum_{k\in p\cap q} w_{kq}^{tfidf} w_{kp}^{tfidf}}{\sqrt{\sum_{k\in p}(w_{kp}^{tfidf})^2 \sum_{k\in q}(w_{kq}^{tfidf})^2}}$$

where q is the topic and w_{kd}^{tfidf} is the *tf*idf* weight of term k in document d:

$$w_{kd}^{tfidf} = f_{kd} \cdot \left(1 + \ln\left(\frac{|S|}{n_k}\right)\right).$$

We do this incrementally as the sets of pages $S(t)$ for each crawler grows with t. This way we can plot a trajectory over time and assess the crawlers based on their capability to remain on topic. In general, this measure assesses the cohesiveness of the retrieved set with the topic as the core. The underlying assumption is that the more cohesive the crawled set, the more relevant its pages.

5. EXPERIMENTAL DESIGN

5.1 Topics

The crawlers considered in this paper are designed for topic oriented page retrieval. Although crawlers related to general-purpose Web search engines are also important, we are not focusing on these.

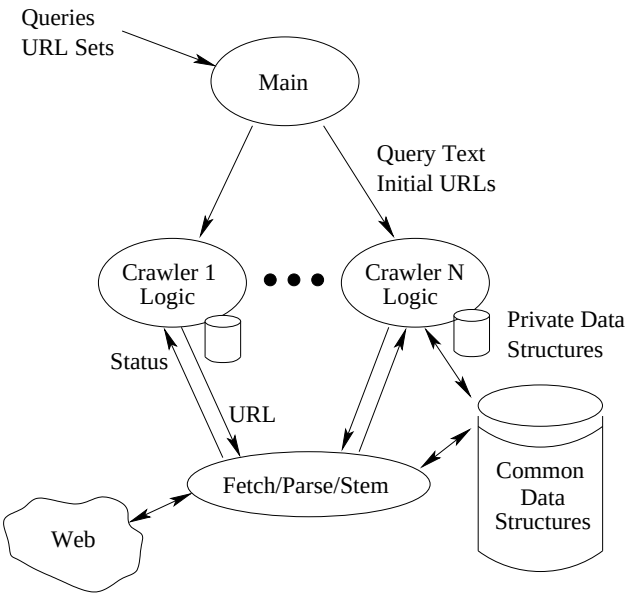

Figure 1: Crawler evaluation architecture.

One major advantage in working with a topic-based rather than a user-based experimental design is that one can essentially use an unlimited number of topics as long as there is a consistent method for generating these topics. The preliminary results reported in this study are based on 100 topics. Although we regard 100 to be a small set, it is still larger than user-based studies such as the one conducted by Amento *et al.* [1], who used 5 queries. Even when compared with other topic-based efforts our topic collection size compares favorably with the work by Cho *et al.* [7], who essentially use 1 topic (the Stanford home page) and by Ben-Shaul *et al.* [2], who discuss 5 topics.

Topics were identified from the Yahoo[1] hierarchy. We first identified all Yahoo "leaves," i.e., pages that have no children category nodes, but rather only external links. These were ordered according to a breadth-first walk. Then we filtered out leaf pages with less then 5 external links. Finally, we selected the first 100 leaf pages to represent the topics. For each topic, we extracted the first 10 external links to be used as seed set. The associated topic was formed by concatenating the textual descriptions of the links in the seed set (both anchor and normal text). Table 1 shows a couple of sample topics.

5.2 Architecture

Figure 1 illustrates the architecture of the system developed to test and evaluate crawler strategies. The system was designed to provide a framework in which modules implementing arbitrary crawling algorithms can be easily plugged in with a standard interface, sharing data structures and utilities to optimize efficiency without affecting the fairness of the evaluation. Examples of common facilities include a cache, an HTTP interface for the Web, a simple HTML parser, a stemmer [18], benchmarking and reporting routines. The system was implemented in Perl.

Each tested crawler can visit up to `MAX_PAGES` = 1,000 pages per topic, starting from the seed set. However, a crawl may end sooner if a crawler's buffer becomes empty as explained below. We use a timeout of 10 seconds for Web downloads. Large pages are chopped so that we retrieve only the first 100 KB. The only protocol allowed is HTTP (with redirection allowed), and we also filter out all but static pages with `text/html` content. Stale links yielding HTTP error codes are removed as they are found (only good links are used in the analysis).

[1] `http://www.yahoo.com`

5.3 Resource Constraints

Crawlers consume resources: network bandwidth to download pages, memory to maintain private data structures in support of their algorithms, CPU to evaluate and select URLs, and disk storage to store the (stemmed) text and links of fetched pages. Obviously the more complex the selection algorithm,the greater the use of such resources. In order to allow for a fair comparison of diverse crawling algorithms, we take two measures:

1. We track the CPU time taken by each crawler for each page and each topic, ignoring the time taken by the fetch module, which is common to all the crawlers. We do this since it is impossible to control for network traffic and congestion, and we want to benchmark only the crawler-specific operations.

2. We limit the memory available to each crawler by constraining its buffer size. This buffer can be used by a crawler to temporarily store pages and links, for example a frontier of pages whose links have not been explored or a subset of the Web used to evaluate URLs. Each crawler is allowed to track a maximum of `MAX_BUFFER` = 200 pages and their links. If the buffer becomes full then the crawler must decide which pages are to be substituted as pages are added.

6. CRAWLERS ASSESSED

6.1 BestFirst

The BestFirst crawler is quite simple. It maintains a frontier of known URLs as a priority queue sorted by the cosine similarity between the topic and the page where the URL was found. Figure 2 offers a simplified pseudo-code description of the BestFirst algorithm. The `sim()` function returns the cosine similarity between topic and page:

$$sim(q,p) = \frac{\sum_{k \in q \cap p} f_{kq} f_{kp}}{\sqrt{\sum_{k \in p} f_{kp}^2 \sum_{k \in q} f_{kq}^2}} \quad (1)$$

where q is the topic and p is the fetched page.

6.2 PageRank

PageRank was proposed by Brin and Page as a possible model of user surfing behavior [4]. The PageRank of a page represents the probability that a random surfer (one who follows links randomly from page to page) will be on that page at any given time. A page's score depends upon the scores of the pages that point to it. Source pages distribute their PageRank across all of their outlinks. Formally:

$$PR(p) = (1-\gamma) + \gamma \sum_{\{d \in in(p)\}} \frac{PR(d)}{|out(d)|} \quad (2)$$

Table 1: Sample topics and seed sets.

Yahoo Category	Topic	Seed Set
/Education /Statistics	Education Census - from the U.S. Census Bureau. National Assessment of Educational Progress - data and reports from the National Center for Education Statistics. National Education Statistical Information Systems (NESIS) - joint programme of UNESCO/ADEA to develop self-sustainable statistical information systems for education policy needs in Africa. School District Data Book Profiles: 1989-1990 - social, financial and administrative data for school districts in the United States. School Enrollment - data from the U.S. Census Bureau.	http://www.census.gov /population/www/socdemo/education.html http://nces.ed.gov/naep/ http://nesis.easynet.fr/ http://govinfo.kerr.orst.edu/sddb-stateis.html http://www.census.gov /population/www/socdemo/school.html
/Arts /Art_History /Criticism_and_Theory	Art Historians' Guide to the Movies - a record of appearances of and references to famous works of painting, sculpture, and architecture in the movies. Art History: A Preliminary Handbook - guide to studying art history. Artists on Art - excerpts from writings and interviews of great artists past and present on the concept and process of art, as well as artist chronologies of the periods in which they worked. Brian Yoder's Art Gallery and Critic's Corner Part - magazine of art and theory produced by CUNY graduate students. Pre-Raphaelite Criticism Underground Art Critic - offering modern art criticism for the postmodern masses.	http://www.rci.rutgers.edu/~eliason/ahgttm.htm http://www.arts.ouc.bc.ca/fiar/hndbkhom.html http://www.constable.net/ http://www.primenet.com/~byoder/art.htm http://Brickhaus.com/amoore/magazine/ http://www.engl.duq.edu/servus/PR_Critic/ http://www.geocities.com/SoHo/Gallery/3431

```
BestFirst(topic, starting_urls) {
   foreach link (starting_urls) {
      enqueue(frontier, link);
   }
   while (#frontier > 0 and visited < MAX_PAGES) {
      link := dequeue_link_with_max_score(frontier);
      doc := fetch_new_document(link);
      score :=  sim(topic, doc);
      foreach outlink (extract_links(doc)) {
         if (#frontier >= MAX_BUFFER) {
            dequeue_link_with_min_score(frontier);
         }
         enqueue(frontier, outlink, score);
      }
   }
}
```

Figure 2: Pseudocode of the BestFirst crawler.

```
PageRank(topic, starting_urls) {
   foreach link (starting_urls) {
      enqueue(frontier, link);
   }
   while (#frontier > 0 and visited < MAX_PAGES) {
      if (multiplies_25(visited)) {
         foreach link (frontier) {
            PR(link) := recompute_PR;
         }
      }
      link := dequeue_link_with_max_PR(frontier);
      doc := fetch_new_document(link);
      score :=  sim(topic, doc);
      if (#buffered_pages >= MAX_BUFFER) {
         dequeue_page_with_min_score(buffered_pages);
      }
      enqueue(buffered_pages, doc);
      foreach outlink (extract_new_links(doc)) {
         if (#frontier >= MAX_BUFFER) {
            dequeue_link_with_min_PR(frontier);
         }
         enqueue(frontier, outlink);
      }
   }
}
```

Figure 3: Pseudocode of the PageRank crawler.

where p is the page being scored, $in(p)$ is the set of pages pointing to p, $out(d)$ is the set of links out of d, and the constant $\gamma < 1$ is a damping factor that represents the probability that the random surfer requests another random page.

As originally proposed PageRank was intended to be used in combination with similarity based criteria to rank retrieved sets of documents. This is in fact how PageRank is used in the Google search engine[2]. However, more recently, PageRank has been used to guide crawlers [7] and to assess page quality [10]. We use a combination of PageRank and similarity with the topic to guide our crawler. At any point during the crawl we select the URL with the highest PageRank as the next URL to traverse. As mentioned before the crawler is allowed to store and utilize the link information of up to `MAX_BUFFER` Web pages in its buffer. When new pages are fetched and the buffer is full, we delete pages that have the least similarity with the topic. For this, similarity is calculated using Equation 1. In addition, we limit the links recorded in the buffer to those that point to domains that are different from the source page domain.

We use an efficient algorithm to calculate PageRank [9]. Even so, as one can see from Equation 2, PageRank requires a recursive calculation until convergence, and thus its computation can be a very resource intensive process. In the ideal situation we would recalculate PageRanks every time a URL needs to be selected from the frontier. Instead, to improve efficiency, we recompute PageRanks only every 25th URL selected. The PageRank crawler algorithm is illustrated in Figure 3. The `sim()` function returns the cosine similarity between topic and page as measured according to Equation 1, and the `PR()` function is computed according to Equation 2. We assume that pages with 0 out-degree (such as the URLs in our crawl frontier) are implicitly linked to every page in our buffered Web, as required for the PageRank algorithm to converge. We use $\gamma = 0.8$ and the threshold for convergence is set at 0.01.

6.3 InfoSpiders

In InfoSpiders [15], a population of agents search for pages relevant to the topic, using evolving query vectors and neu-

[2] `http://www.google.com`

```
InfoSpiders(topic, starting_urls) {
   foreach agent (1 .. #starting_urls) {
      initialize(agent, query);
      situate(agent, starting_urls);
      agent.energy := THETA / 2;
   }
   cost := #starting_urls * THETA / (2 * MAX_PAGES);
   while (pop_size > 0 and visited < MAX_PAGES) {
      agent := pick_random_agent;
      agent.link := pick_link_from_current_doc(agent);
      agent.doc := fetch_new_document(agent);
      agent.energy +=  sim(topic, agent.doc) - cost;
      nnet_learn_to_predict(agent, sim(topic, agent.doc));
      if (agent.energy >= THETA and pop_size < MAX_BUFFER) {
         offspring := mutate(clone(agent));
         offspring.energy := agent.energy / 2;
         agent.energy -= offspring.energy;
      }
      elseif (agent.energy <= 0) kill(agent);
   }
}
```

Figure 4: Pseudocode of the InfoSpiders crawler.

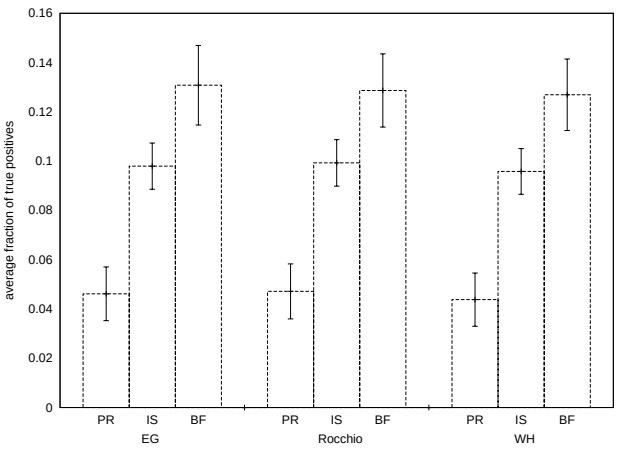

Figure 5: Assessment of crawlers using classifiers. Legend: PR=PageRank, IS=InfoSpiders, BF=BestFirst. Error bars in this and the following plots correspond to standard errors.

ral nets to decide which links to follow. Fig. 4 illustrates our simplified implementation of the original algorithm as a crawler module. This evolutionary approach uses a fitness measure based on similarity as a local selection criterion.

The adaptive representation of an agent consists of a vector of keywords (initialized with the topic terms) and a neural net used to evaluate links. We used a simple perceptron in the crawler implemented here, represented as a vector of real-valued weights. The keywords represent an agent's opinion of what terms best discriminate good pages from the rest, and the network weights model the relative importance of such terms. The neural net has a real-valued input for each keyword and a single output unit.

In the main loop of Fig. 4, an agent estimates the links in the current page by considering the text surrounding those links. This is done by feeding into the agent's neural net activity corresponding to the keywords to which it is sensitive. Each input unit of the neural net receives a weighted count of the frequency with which the keyword occurs in the vicinity of the link to be traversed. A distance weighting function is used, which is biased towards keyword matches closest to the link in question. Once the process is repeated for each link in the current document, the agent uses a stochastic selector to select one of the links.[3] This is different from BestFirst where all links from a page are considered equal.

After the new document has been fetched, its cosine similarity to the agent's query vector is computed according to Equation 1 and compared with the estimate made from the previous page, in order to improve on the agent's prediction. The difference is used to adjust the neural net's weights using standard back-propagation[4] [19].

The agent's "energy" is then updated. Agents are rewarded with energy in proportion to the similarity between the topic and the newly fetched documents, and charged with a constant cost for each fetched page.[5] An agent's energy level is used to determine whether the agent should die, reproduce, or survive. An agent dies when it runs out of energy. The energy level is compared with a constant reproduction threshold `THETA` $= 2$ to determine reproduction. At reproduction, the agent's offspring receives half of its parent's energy. The offspring also has its keyword vector mutated by replacing the term that is least frequent in the current document with a term that is most frequent. This mutation provides InfoSpiders with the unique capability to adapt its search strategy based on new clues captured from promising pages.

7. RESULTS

Figure 5 shows the results for our first evaluation method, using each of the three classifier techniques — Rocchio, WH and EG. In Section 4.1 we described the training set for a classifier. Figure 5 presents the results of classifying each set of $1,000$ crawled pages using the trained classifiers. The Y axis represents the average fraction of crawled pages that were classified as positive by the topic's classifier. The graph presents means and standard errors (across topics) for the three crawlers for each classifier. It shows that BestFirst outperforms InfoSpiders, which in turn outperforms PageRank. This trend is consistent across classifiers. Moreover a t-test analysis reveals that all pairwise differences between crawlers within each classifier are statistically significant at the 95% confidence level, which is consistent with the error bars. As mentioned before this measure in a sense evaluates crawler precision where content-based relevance is decided by the trained classifier. Viewed in this way, the best classifier results are close to 13% precision. The low precision may in part be due to the fact that there were very few positive example pages (5 to 10 per topic) compared to negative example pages (around 700 on average) for training the classifiers. Thus the classifiers may tend towards making more negative decisions.

Figure 6 presents the results using our second evaluation method with SMART as the independent retrieval system. The X axis of the graph represents sequence `id`'s for the fetched pages (1 through 1,000). The Y axis represents the mean SMART rank for pages fetched up to the sequence position on the X axis. These RankScores are also averaged

[3]We use a Boltzmann distribution with temperature 0.125.
[4]We use 5 epochs of training with learning rate 0.5 and momentum 0.9.
[5]If a page is visited again, no energy is gained and the cost is reduced by a factor of 0.1.

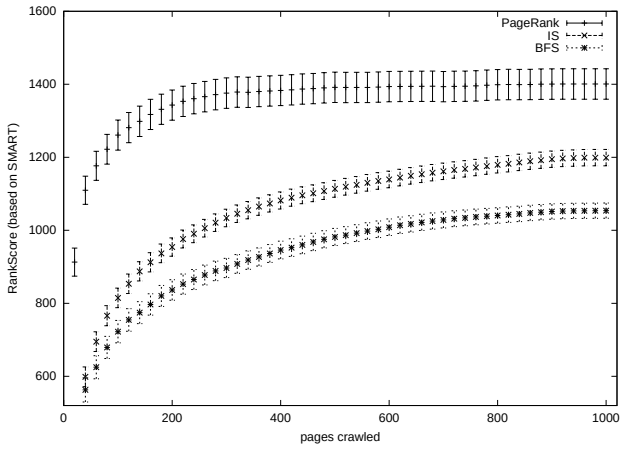

Figure 6: Assessment of crawlers using RankScore derived from SMART-based rankings.

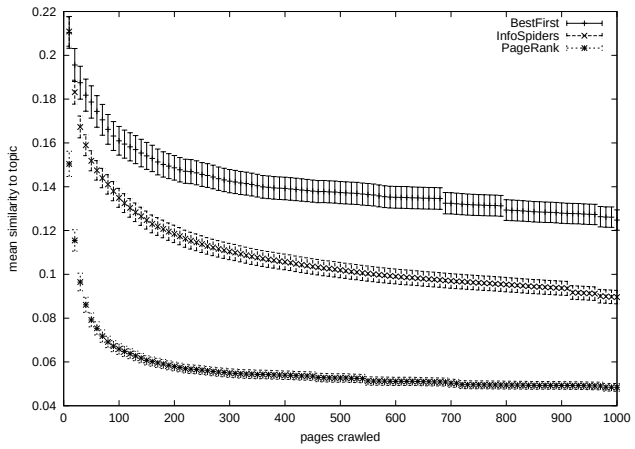

Figure 7: Assessment of crawlers using mean similarity between topics and crawled pages.

across topics for each crawler, yielding standard errors. Recall that lower RankScore indicates better performance. The graph shows that the best average RankScores are achieved by BestFirst, and the worst by PageRank. Differences are statistically significant except between BestFirst and InfoSpiders at the very beginning of the crawls. In a sense the RankScore measure combines the notions of recall and average search length.

Figure 7 shows the results from our last evaluation strategy. Here we measure the similarity maintained between the set of pages visited by a crawler and the corresponding topic. These similarity scores are averaged across topics for each crawler, yielding standard errors. The figure offers a temporal view since the X axis represents time as measured by crawl size. Again, very soon in the crawl a significant advantage is established by BestFirst over InfoSpiders, and by the latter over PageRank.

8. CONCLUSIONS

We have introduced three methods and related metrics to evaluate the performance of topic-driven crawler algorithms. We have implemented three crawlers based on algorithms suggested in the literature, and compared them using the proposed evaluation techniques.

We are pleased that the three evaluation methods seem to yield consistent results. This is not entirely surprising, given that they are all based on similarity measures of some kind. We should stress however that the classifier-based method is not entirely driven by similarity, since the negative examples used for building the classifiers are drawn from pages independent of the topic. The method based on a retrieval system and the one based on mean similarity to topic give us quite informative data on the dynamics of the crawler, which is an important feature given the contrast between the dynamic process of crawling versus static retrieval.

The assessments based on a retrieval system could be augmented to add to the confidence in this method, by using more than one system to assess page ranks. For example we could use both the vector based SMART system and the boolean search interface of MG [24] to provide an average rank for each page in the pool of fetched pages, and use these averages to rate the crawlers.

The analyses outlined in this paper are all based on averages over topics. Further work is needed to analyze topic-specific performance.

Other criteria beyond similarity to the topic might be important for a crawler. While the experiments described here are limited to the task of locating pages similar to the topic, one might construct metrics that focus on factors such as novelty, recency, or authority. For example we expect InfoSpiders to have a potential advantage in exploiting novelty because of its ability to expand the query with new terms that appear to be correlated with energy. This potential is not captured in the evaluation methods proposed in this paper. Likewise, PageRank would arguably have an advantage in locating authoritative pages.

One limiting assumption that we have made with respect to all the proposed evaluation methods is that no user assessments are available and therefore the relevant set for each topic is unknown. If one had knowledge of the relevant set, several useful metrics would become available, such as precision and recall. Clearly search length (the number of pages a crawler has to visit to get various percentages of the relevant set) would be particularly appropriate to evaluate crawling strategies [15]. However it is difficult to imagine that in a dynamic environment such as the Web anyone might have access to the exhaustive knowledge necessary to determine a topic's entire relevant set. Therefore we have focused on metrics that do not require such knowledge.

Regarding the comparison between the three tested crawlers, the results are consistent in showing that BestFirst outperformed the other two, and PageRank finished last. The success of BestFirst is not surprising for a topic-driven crawler since similarity to the topic is the only criterion used to prioritize over links.

The poor performance of PageRank was also to be expected since this algorithm favors authoritative pages in a general context, without regard to the topic in question. So the crawler rapidly drifts away as shown in Figure 7, often attracted by popular but irrelevant sites such as `netscape.com` and `real.com`. As implemented, PageRank appears too general for the topic-driven task.

The performance of InfoSpiders was somewhat disappointing, because the agents in this crawler can use similarity to converge to a relevant area of the Web, and furthermore use

their neural nets to discriminate among links from a page — something BestFirst cannot do. It has been suggested that InfoSpiders could complement search engines to locate highly relevant pages that are too recent and/or distant to have been indexed by other crawlers [15]. To test this, we considered the set of 50 best pages as ranked by SMART, and found that InfoSpiders crawled $(62 \pm 2)\%$ of such set, versus $(52 \pm 3)\%$ and $(12 \pm 1)\%$ by BestFirst and PageRank, respectively. This is consistent with InfoSpiders' promise to achieve the highest coverage of the most relevant pages.

In the future we intend to study ways to evaluate InfoSpiders' capability to locate novel relevant pages with small similarity to the query. One way would be to measure similarity between visited pages and an expanded query made of pages that are known to be related to the topic. This would also allow us to experiment with more realistic (shorter) queries in place of the topics used here.

It should also be noted that our implementation of InfoSpiders left out several aspects of the original model for the sake of simplification and efficiency. For example we use a perceptron rather than a multi-layer neural network, we tune its weights by 1-step prediction rather than by Q-learning, and we perform very few epochs with a high learning rate. The size of the InfoSpiders population is also kept significantly below the `MAX_BUFFER` limit and thus this crawler does not fully exploit the memory resources available to the other two. Our architecture will allow us to test a more complete implementation of the algorithm in future research.

The number of queries used in the experiments described here was limited by the fact that we had to go through several iterations of software design in order to decrease the time taken by the crawlers from an average of 7 hours (clock time) for a topic to between 0.5 and 3 hours. We intend to extend our experiments to much larger query sets, and will be able to do so thanks to the infrastructure we have established in building our evaluation system.

The evaluation system can also be used to assess many alternative crawling strategies; we intend to test various variations of PageRank (e.g., using different convergence criteria and recomputing PageRanks at different frequencies) and InfoSpiders (e.g., using lookahead and different learning algorithms) as well as other crawlers. The scalability and sensitivity of each crawler's performance should also be gauged versus its available resources (i.e. the size of the buffer data structures and the length of the crawl).

Finally, time is an important factor we have neglected in the preliminary results reported here. The results assume that each crawler module has the same time complexity in determining the next page to be visited. Clearly this is not the case since different crawling algorithms have different computational requirements. Therefore we have monitored the CPU time required by each crawler module to pick each link. We intend to use this data to carry out a quality/cost analysis, normalizing the assessment of each crawler's performance by its time complexity.

9. ACKNOWLEDGMENTS

We are grateful to Micah Wedermeyer for implementing an early version of the PageRank algorithm.

10. REFERENCES

[1] B. Amento, L. Terveen, and W. Hill. Does "authority" mean quality? predicting expert quality ratings of web documents. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 296–303, 2000.

[2] I. Ben-Shaul, M. Herscovici, M. Jacovi, Y. Maarek, D. Pelleg, M. Shtalhaim, V. Soroka, and S. Ur. Adding support for dynamic and focused search with fetuccino. In *Proceedings of 8th International World Wide Web Conference*, 1999.

[3] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, 1998.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia*, 1998.

[5] 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 7th International World Wide Web Conference*, 1998.

[6] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *Proceedings of 8th International World Wide Web Conference*, 1999.

[7] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. In *Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia*, 1998.

[8] P. De Bra 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*, 1994.

[9] T. Haveliwala. Efficient computation of pagerank. Technical report, Stanford Database Group, 1999.

[10] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. Measuring search engine quality using random walks on the web. In *Proceedings of 8th International World Wide Web Conference*, pages 213–225, 1999.

[11] D. A. Hull. Improving text retrieval for the routing problem using lat ent semantic indexing. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 282–289, Dublin, IE, 1994. Springer Verlag, Heidelberg, DE.

[12] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear p redictors. Technical Report Technical Report UCSC-CRL-94-16, Baking Center for Computer Engineering & Information Scien ces; University of California, Santa Cruz, CA, 1994.

[13] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[14] D. Lewis, R. Schapire, J. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–303, 1996.

[15] F. Menczer and R. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. *Machine Learning*, 39(2/3):203–242, 2000.

[16] H. Ng, W. Goh, and K. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 67–73, 1997.

[17] B. Pinkerton. Finding what people want: Experiences with the webcrawler. In *Proceedings of the First International World Wide Web Conference, Geneva, Switzerland*, 1994.

[18] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. http://www.muscat.com/~martin/stem.html.

[19] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, chapter 8. Bradford Books (MIT Press), Cambridge, MA, 1986.

[20] G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[21] I. Silva, B. Ribeiro-Neto, P. Calado, N. Ziviani, and E. Moura. Link-based and content-based evidential information in a belief network model. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, 2000.

[22] C. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979. Second edition.

[23] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[24] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, CA, 1999. Second Edition.

[25] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1):69–90, 1999.