A General Evaluation Framework for Topical Crawlers

P. Srinivasan (padmini-srinivasan@uiowa.edu)*
School of Library & Information Science and Department of Management Sciences, University of Iowa, Iowa City, IA 52242

F. Menczer (fil@indiana.edu)†
School of Informatics and Department of Computer Science, Indiana University, Bloomington, IN 47408

G. Pant (gautam-pant@uiowa.edu)
Department of Management Sciences, University of Iowa, Iowa City, IA 52242

Abstract. Topical crawlers are becoming important tools to support applications such as specialized Web portals, online searching, and competitive intelligence. As the Web mining field matures, the disparate crawling strategies proposed in the literature will have to be evaluated and compared on common tasks through well-defined performance measures. This paper presents a general framework to evaluate topical crawlers. We identify a class of tasks that model crawling applications of different nature and difficulty. We then introduce a set of performance measures for fair comparative evaluations of crawlers along several dimensions including generalized notions of precision, recall, and efficiency that are appropriate and practical for the Web. The framework relies on independent relevance judgements compiled by human editors and available from public directories. Two sources of evidence are proposed to assess crawled pages, capturing different relevance criteria. Finally we introduce a set of topic characterizations to analyze the variability in crawling effectiveness across topics. The proposed evaluation framework synthesizes a number of methodologies in the topical crawlers literature and many lessons learned from several studies conducted by our group. The general framework is described in detail and then illustrated in practice by a case study that evaluates four public crawling algorithms. We found that the proposed framework is effective at evaluating, comparing, differentiating and interpreting the performance of the four crawlers. For example, we found the IS crawler to be most sensitive to the popularity of topics.

Keywords: Web crawlers, evaluation, tasks, topics, precision, recall, efficiency.

1. Introduction

Topical crawlers, also known as topic driven or focused crawlers, are an important class of crawler programs that complement search engines. Search engines serve the general population of Web users. In contrast, topical crawlers are activated in response to particular information needs. These could be from an individual user (query time or online

* Contact author. Tel: +1-319-335-5708, Fax: +1-319-335-5374.

† Partially supported by National Science Foundation CAREER grant No. IIS-0133124/0348940

crawlers) or from a community with shared interests (topical or vertical search engines and portals). The crawlers underlying search engines are designed to fetch as comprehensive a snapshot of the Web as is possible; topical crawlers are designed to target portions of the Web that are relevant to the triggering topic. Such crawlers have the advantage that they may in fact be driven by a rich context (topics, queries, user profiles) within which to interpret pages and select the links to visit. Today, topical crawlers have become the basis for many specialized services such as investment portals, competitive intelligence tools, and scientific paper repositories.

Starting with the early breadth first, exhaustive crawlers (Pinkerton, 1994) and depth first crawlers such as Fish Search (De Bra and Post, 1994) defining the beginnings of this research area, we now see a variety of crawling algorithms. There is Shark Search (Hersovici et al., 1998), a more aggressive variant of De Bra's Fish Search. There are crawlers whose decisions rely heavily on link based criteria (Cho et al., 1998). Others, such as the Focused Crawler (Chakrabarti et al., 1999), exploit the lexical and conceptual knowledge provided by a topic hierarchy. Still others, such as InfoSpiders (Menczer and Belew, 2000), emphasize contextual knowledge for the topic (Hersovici et al., 1998; Menczer and Belew, 1998; Aggarwal et al., 2001; Chakrabarti et al., 2002b) including that acquired by experience (Menczer and Belew, 1998), by reinforcement learning (Menczer, 1997; Rennie and McCallum, 1999), or by relevance feedback (Menczer and Belew, 2000). In a companion paper we study several machine learning issues related to crawler algorithms, including for example, the role of adaptation in crawling and the scalability of algorithms (Menczer et al., 2004).

One research issue that is gathering increasing momentum is the evaluation of topical crawlers. The rich legacy of information retrieval research comparing retrieval algorithms in the non-Web context offers many evaluation methods and measures that may be applied toward 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.

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. Conducting such user-based evaluations of Web crawlers is very challenging. 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. The number of documents to be

assessed is also very large compared to traditional information retrieval systems. Crawlers typically visit between 10^4 and 10^7 pages.

Crawls against the live Web also 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 while keeping user based evaluation results as the ideal, we explore alternative 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.

Our analysis of the Web information retrieval literature (Aggarwal et al., 2001; Amento et al., 2000; Ben-Shaul et al., 1999a; Bharat and Henzinger, 1998; Chakrabarti et al., 1998; Chakrabarti et al., 1999; Chakrabarti et al., 2002b; Henzinger et al., 1999; Hersovici et al., 1998; Najork and Wiener, 2001; Silva et al., 2000) and our own experience (Menczer, 1997; Menczer and Belew, 1998; Menczer and Belew, 2000; Menczer et al., 2001; Menczer, 2003; Pant and Menczer, 2002; Pant et al., 2002; Menczer et al., 2004) indicate that in general, when embarking upon an experiment comparing crawling algorithms, several critical decisions are made. These impact not only the immediate outcome and value of the study but also the ability to make comparisons with future crawler evaluations. In this paper we offer a general framework for crawler evaluation research that is founded upon these decisions. Our goal is both to present this framework and demonstrate its application to the evaluation of four off-the-shelf crawlers. Our generic framework has three distinct dimensions. The first dimension is regarding the nature of the crawl task addressed (Section 2). This includes consideration of how topics are defined and how seeds and target relevant pages are identified. The second dimension deals with evaluation metrics both for effectiveness and for efficiency analysis (Section 3). The last dimension of the framework looks at topics in greater detail, by examining particular characteristics such as popularity and authoritativeness and their effect on crawler behavior (Section 4). We present this framework as a means for systematically increasing our understanding of crawler technologies through experimentation. After

these sections, we take four off-the-shelf crawlers and compare them using this framework (Section 5). We conclude in Section 6 with a discussion on the experiment in our case study and on the evaluation framework in general.

2. Nature of Crawl Tasks

A crawl task is characterized by several features. These include how the topic is defined, the mechanism by which seed pages for starting the crawl are selected and the location of the topic's relevant target pages[1] relative to the seed pages. Obviously, a crawl task where the seeds link directly to many pages relevant to the topic is likely to be less challenging than one in which the seeds and targets are separated by some non trivial link distance. These issues are discussed in this section.

2.1. Topics and Descriptions

Unlike questions that are built around inquiries of some kind, a topic such as 'sports' or 'US Opens' or 'anthrax' delineates a particular domain of discourse. As seen in various examples (Aggarwal et al., 2001; Ben-Shaul et al., 1999b; Bharat and Henzinger, 1998; Hersovici et al., 1998; Chakrabarti et al., 1999; Chakrabarti et al., 2002a), topics offer a handy mechanism for evaluating crawlers, since we may examine their ability to retrieve pages that are on topic. Topics may be obtained from different sources as for instance asking users to specify them. One approach is to derive topics from a hierarchical index of concepts such as Yahoo or the Open Directory Project (ODP) (Chakrabarti et al., 1999; Menczer et al., 2001; Pant et al., 2002). A key point to note is that all topics are not equal. Topics such as '2002 US Opens' and 'trade embargo' are much more specific than 'sports' and 'business' respectively. Moreover, a given topic may be defined in several different ways, as we describe below.

Topic specification plays a critical role in our framework. We start by asking: Given a hierarchy of concepts how are topics to be specified? One method is to use the leaf node concepts as topics (Menczer et al., 2001). The problem with this approach is that the selected topics may be at different levels of specificity. In our framework we control for

[1] Henceforth we use the term 'relevant pages' and relevance in general to refer to topical relevance and not to end user based relevance judgments. We recognize that there is a significant difference between the two with several factors such as recency influencing the latter alone.

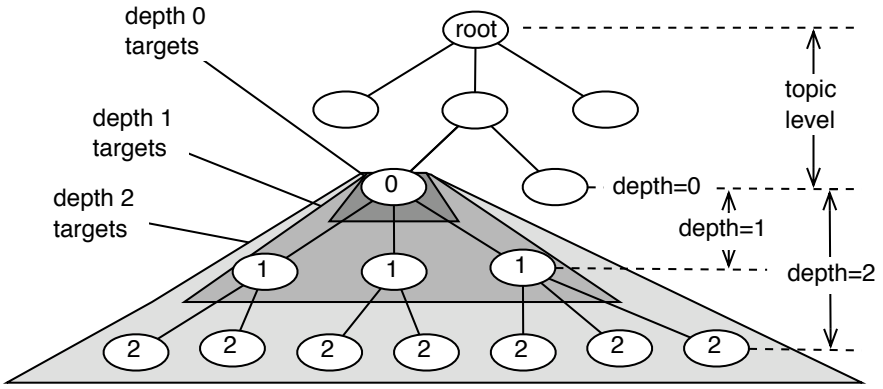

Figure 1. Illustration of a topic subtree from a hierarchical directory. The topic in this example has `TOPIC_LEVEL`=2 and `MAX_DEPTH`=2. Topic nodes are labeled with their depth. The external pages linked from nodes at a given depth are the targets for that depth. Shaded areas represent target sets corresponding to subtrees of depth between 0 and `MAX_DEPTH`, i.e., to progressively broader interpretations of the topic. A broader interpretation (lighter shade of gray) includes additional, more specific targets.

this by deriving topics from concept nodes that are at a predefined distance (`TOPIC_LEVEL`) from the root of the concept hierarchy. While one cannot say that two topics at the same level in the ODP hierarchy have the same specificity, it is reasonable to assume that `TOPIC_LEVEL` is correlated with specificity, and it is useful to have a simple control parameter to characterize the specificity of a set of topics. Once a topic node is identified, the topic *keywords* are formed by concatenating the node labels from the root of the directory tree to the topic node. These keywords form the search criteria provided as initial input to the crawlers.

Instead of building topics from single nodes we take a more general approach and build them from subtrees of a given maximum depth (`MAX_DEPTH`) whose roots are `TOPIC_LEVEL` links away from the root of the original concept tree. Depth, as used here, refers to the height of a subtree. Figure 1 illustrates these ideas with a topic subtree of `MAX_DEPTH` = 2 built from a concept hierarchy at `TOPIC_LEVEL` = 2.

In our framework, subtrees offer a systematic way to delineate topics. Moreover, by varying a parameter `DEPTH` from 0 to `MAX_DEPTH`, it is possible to generate alternative *descriptions* of a given topic. These descriptions may be used to estimate relevance of retrieved pages. If we use information from the root of the topic subtree alone (`DEPTH` = 0), then we get the most minimal set of topic descriptions. Both the descriptive text that embeds the external links and the anchor text that

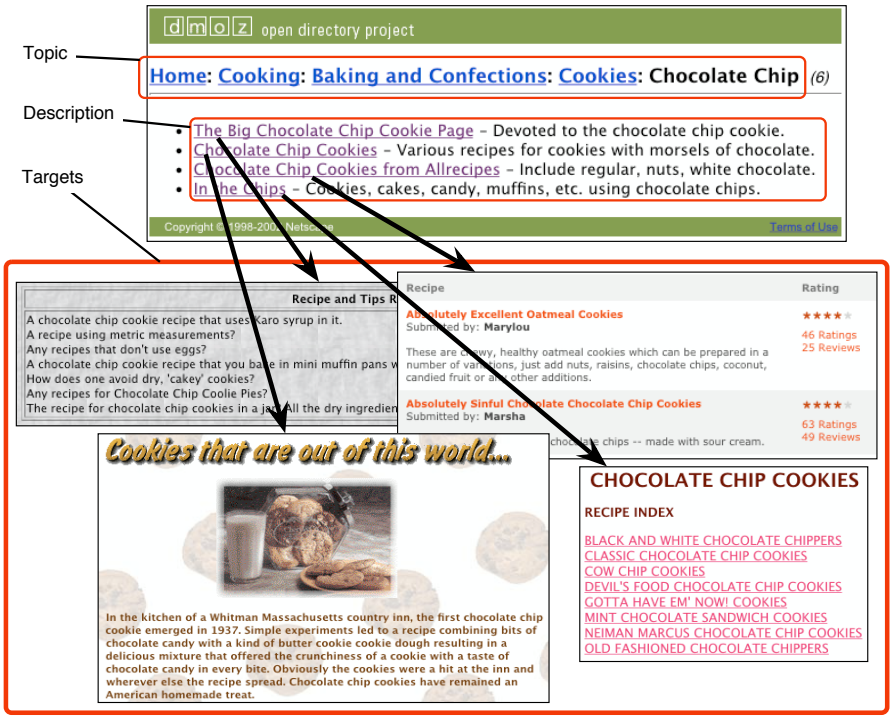

Figure 2. Illustration of a topic node from the Open Directory (`dmoz.org`), with its associated topic keywords, description, and target set. Note that the keywords correspond to category labels along the path from the ODP root to the topic node. In this abridged example the path has 5 edges, therefore `TOPIC_LEVEL`=5. Since the topic in this example is a leaf node (no subtopics), the only possible target set corresponds to `DEPTH`=0.

labels the external links in the page at the root of the topic subtree may be used to provide the minimal description of the topic. Note that in a manually edited directory such as ODP, the textual descriptions of external pages are written by human experts, independent of the authors who generate the content of the pages described. If in addition, we use information from the next level of nodes in the subtree (`DEPTH` = 1), then we get a more detailed view of the topic and so on till the leaf nodes of the subtree are involved (`DEPTH` = `MAX_DEPTH`). Thus a single topic may have `MAX_DEPTH` + 1 sets of descriptions that differ in their level of detail. Descriptions at higher depths include those at lower depths. Figure 2 illustrates the concept of topic description with an example corresponding to a leaf topic, i.e., `DEPTH` = `MAX_DEPTH` = 0.

2.2. Target Pages

Hierarchical concept based directories are designed to assist the user by offering entry points to a set of conceptually organized Web pages. Thus the Yahoo directory page on *Newspapers* leads to *USA Today*, *New York Times* and the Web sites of other news media. In effect, one may regard the resources pointed to by the external links as the topically relevant target set for the concept represented by the directory page: *USA Today* and *New York Times* may be viewed as part of the set of target relevant pages for the concept of Newspapers.

In our framework, parallel to topic descriptions, topic target pages are also differentiated by the `DEPTH` of the topic subtree. Thus when the topic is described by a subtree of `DEPTH` = 0, then the relevant target set consists of the external links from the root node of the topic subtree. Such an example is depicted in Figure 2. The target set corresponding to the topic description at `DEPTH` = 1 also includes the external links from the topic's nodes at this level and so on. Thus for a single topic there are `MAX_DEPTH` + 1 sets of target pages defined, with the set at a higher depth including the sets at the lower depths.

2.3. Seed Pages

The specification of seed pages is one of the most crucial aspects defining the crawl task. The approach used in several papers (Chakrabarti et al., 1999; Ben-Shaul et al., 1999a; Menczer et al., 2001; Menczer, 2003; Pant and Menczer, 2002) is to start the crawlers from pages that are assumed to be relevant. In other words some of the target pages are selected to form the seeds. This type of crawl task mimics the query by example search mode where the user provides a sample relevant page as a starting point for the crawl. These relevant seeds may also be obtained from a search engine (Pant and Menczer, 2002; Srinivasan et al., 2002). The idea is to see if the crawlers are able to find other target pages for the topic. An assumption implicit in this crawl task is that pages that are relevant tend to be neighbors of each other (Menczer, 1997; Davison, 2000; Menczer, 2004). Thus the objective of the crawler is to stay focused, i.e., remain within the neighborhood in which relevant documents have been found.

An alternative way to choose seed pages allows one to define crawl tasks of increasing difficulty. If the seeds are distant from the target pages, there is less prior information available about the target pages when the crawl begins. Links become important not only in terms of the particular pages pointed to, but also in terms of the likelihood of reaching relevant documents further down the path (Diligenti et al., 2000). This problem is also very realistic since quite commonly users

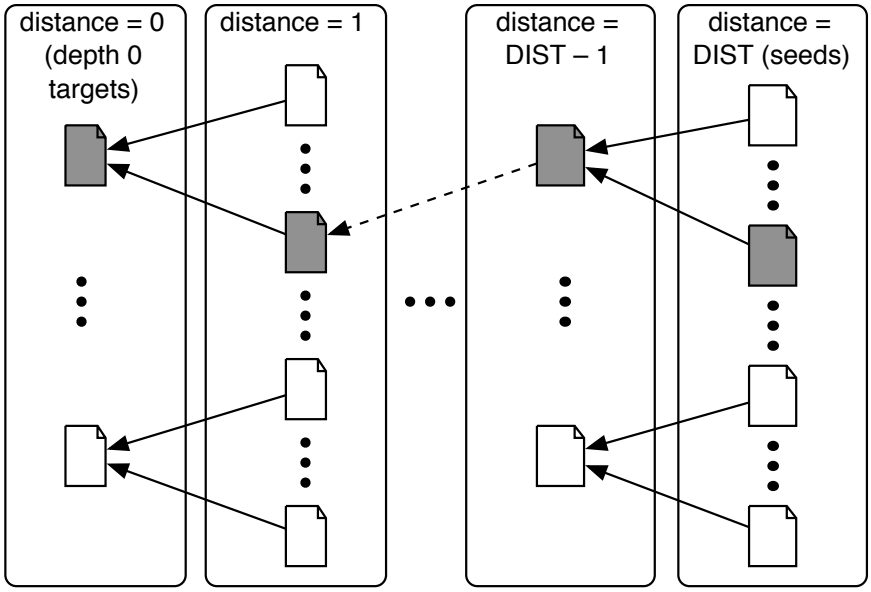

Figure 3. Illustration of the procedure to select seed pages starting from `DEPTH` = 0 targets and moving `DIST` links backwards. For `DIST` > 0, increasing `DIST` makes the crawl task more and more difficult (see text).

are unable to specify a known relevant page and also the search engines may not return relevant pages. With a few exceptions, this class of tasks is rarely considered in the literature. The effort by Aggarwal et al. (2001) is somewhat related in that the authors start the crawl from general points such as Amazon.com. Although Cho et al. (1998) start their crawls at a general point, i.e., the Stanford Web site, topics have rather primitive roles in their research.

Our framework takes a general approach and provides a mechanism to control the level of difficulty of the crawl task. One may specify a distance `DIST` = $1, 2, \ldots$ links between seeds and targets. As `DIST` increases, so does the challenge faced by the crawlers. The procedure in Figure 4 implements the selection of up to `N_SEEDS` seed pages for a given topic.

We start from the `DEPTH` = 0 target pages and select a set of seed pages such that there is a forward path of `DIST` from each seed to some target. The `get_backlinks()` function submits to a search engine a `link:` query for each URL in the set identified by its first argument. The search engine returns a set of backlinks, i.e. URLs of pages that link to the URL in the query. These backlinks form a new set of pages to be sampled in the next iteration of the loop. The `constrained_random_subset()` function guarantees that at most one

```
select_seeds (DIST, N_SEEDS, N_TOPICS, N_QUERIES) {
n_sample = N_QUERIES / (N_TOPICS * DIST);
  seed_set = targets(DEPTH = 0);
  repeat DIST times {
    sample = constrained_random_subset(seed_set, n_sample);
    seed_set = get_backlinks(sample);
  }
  return constrained_random_subset(seed_set, N_SEEDS);
}
```

Figure 4. Pseudocode for the seed page selection procedure.

URL is (randomly) selected among the backlinks of each page from the previous iteration. The procedure may return fewer than N_SEEDS pages because the set in the final iteration may not contain sufficient URLs. Note that the n_sample variable is set to ensure that over all topics, the search engine is queried exactly N_QUERIES times. If there is no practical limitation on the number of search engine queries, N_QUERIES can be set high enough to make n_sample equal to the size of the DEPTH = 0 target set. That way all DEPTH = 0 targets will be reachable from the seed set. Otherwise n_sample targets will be reachable from the seeds. More precisely, we guarantee that in the absence of broken links there exists a path with a minimum length of at most DIST links from each seed page to one of n_sample target pages at DEPTH = 0. The procedure for selecting seed pages is illustrated in Figure 3.

Observe that a breadth first crawler would have a chance in $\ell^{\texttt{DIST}}$ to visit a target from a seed, assuming an average fanout of ℓ links. Therefore a crawler starting from N_SEEDS pages will find a target at most DIST links away by crawling $\texttt{N\_SEEDS} \cdot \ell^{\texttt{DIST}}$ pages. For $\ell \approx 7$ (Kumar et al., 2000), N_SEEDS ≈ 10 and DIST=2, it would be reasonable to expect a non-zero recall of target pages after crawling N_PAGES $\approx 10 \cdot 7^2 = 490$ pages. Larger DIST values correspond to more difficult tasks in the sense that a breadth first crawler would have to visit more pages to find some targets, and conversely smaller DIST values correspond to easier tasks.

The strategy of seeding the crawl at target pages, described earlier, corresponds to a special case DIST = 0 in which the above analysis does not hold. If all of the known targets are used as seeds, we cannot use a crawler's capability to locate target pages as a measure of its performance. If only a subset of the known targets is used to form the seed set, locating the remaining targets can still be used to gauge performance; however we cannot compare the difficulty of such a task

with the DIST > 0 cases because we cannot estimate the link distance between the seeds and the remaining targets.

In summary, as regards crawl task definition, our framework capitalizes on the structure of hierarchical Web directories. Topics are defined as subtrees of such a hierarchy with a TOPIC_LEVEL parameter to control for specificity of topics. Note that the approach of using leaf nodes as topics is a special case of the above with maximal TOPIC_LEVEL and MAX_DEPTH $= 0$. Alternative topic descriptions, varying in extent of detail, may be derived by considering different regions of the topic subtree via the DEPTH parameter. Target sets of relevant pages are also identifiable for each DEPTH as the external resources linked by the directory pages. When it comes to seed pages, since the subtree really represents a single topic, a single set of seed pages is identified for each topic. This is done by starting an iterative backlink procedure from the target pages at the root of the topic subtree (DEPTH $= 0$). Seed pages are chosen such that, barring broken links, from each seed there is at least one target page at most DIST links away. By varying DIST we can evaluate crawlers on problems of varying difficulty. Finally, keywords for a topic are selected by concatenating node labels from the root to the topic node.

3. Crawler Evaluation Metrics

The second major dimension of our general evaluation framework is regarding the evaluation measures required to assess crawlers. In the previous section we discussed how relevant target sets of pages may be identified. These relevant sets provide a convenient basis for computing crawler specific recall and precision scores. But the question still remains: How does one gauge the topical relevance of new pages, i.e., pages that are retrieved but not in the target set? Although target sets are very useful for evaluation, they have been defined using a local strategy, i.e., by exploiting the direct links from the directory pages. Thus we need measures to gauge the relevance of new pages that are retrieved.

A second aspect that needs to be addressed is the following. Assuming that we have a mechanism for assessing page relevance, we need to be able to summarize in some reasonable way the performance of a crawler. In an ideal world one would appreciate having a single number or score such that differences in scores indicate differences in value of the crawlers. However, generating a single number such as recall, precision or an F-score (van Rijsbergen, 1979) is complicated by the fact that crawlers have a temporal dimension. Depending upon the

situation, performance may need to be determined at different points of the crawl. A person interested in quickly obtaining a few relevant pages wants crawlers that return speedy dividends. For a crawler operating to establish a portal on behalf of a community of users, both high recall and high precision are critical after a reasonably large crawl span. These are some of the issues that must be considered when deciding on methods to summarize crawler performance. Accordingly, this section discusses strategies for gauging the importance of new pages not in the target set as well as methods for summarizing crawler performance.

3.1. Background

Page relevance measures that have been considered in the literature are generally of two types: those that use lexical criteria and those that use link based criteria. Lexical measures show a range of sophistication. Cho et al. (1998) explore a rather simple 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. (2000) compute similarity between a page's vector and the centroid of the seed documents as one of their measures of page quality. Chakrabarti et al. (1999) apply classifiers built using positive and negative example pages to determine page importance. Aggarwal et al. (2001) adopt a more generic framework that allows for user designed plug-in modules specifying relevance criteria. The modules that they use in their tests require the presence of pre-specified words in particular parts of the page, such as the URL. Similarity to the topic measured using page text (Bharat and Henzinger, 1998) or the words surrounding a link (Chakrabarti et al., 1998) may also be used to augment what are primarily link based relevance measures.

In-degree, out-degree, PageRank (Brin and Page, 1998), hubs and authorities are commonly used link based page importance measures (Amento et al., 2000; Ben-Shaul et al., 1999b; Bharat and Henzinger, 1998; Chakrabarti et al., 1998; Chakrabarti et al., 1999; Cho et al., 1998). Cho et al. (1998) consider pages with PageRank above a specified threshold as being relevant to the query. Kleinberg's (1999) recursive notion of hubs and authorities has been extended by several others. For example, edge weights are considered important (Chakrabarti et al., 1999) and so are edges that connect different sites (Amento et al., 2000; Bharat and Henzinger, 1998; Chakrabarti et al., 1999). Link based quality metrics rely on the generally reasonable notion of links reflecting the credibility of the target pages. Amento et al. (2000) show that in-degree, authority and PageRank are effective at identifying high quality pages as judged by human experts.

The literature shows a wide variety of summarization methods. The following are just a sample. Given a particular measure of page importance Cho et al. (1998) examine the percentage of important pages retrieved over the progress of the crawl. Menczer et al. (2000) measure search length, i.e., the number of pages crawled until some predetermined fraction of important pages have been visited. Chakrabarti et al. (1999; 2002b) compute the average "harvest rate," which is a running average, over different time slices of the crawl, of page relevance assessed using classifiers. Aggarwal et al. (2001) also use harvest rate, similarly defined as the rate at which crawled pages satisfy a given predicate; if a classifier is used to give numeric relevance values then a page is said to satisfy a predicate if the relevance value exceeds a certain threshold. Najork and Weiner (2001) plot the average day on which the top N pages are retrieved, where N is a variable. Diligenti et al. (2000) examine the average relevance of pages computed using a sliding window of 200 downloads. Rennie and McCallum (1999) compute the percentage of relevant pages found. Finally in our own research we have examined the average rank of the retrieved pages over the progress of the crawl (Menczer et al., 2001).

3.2. Effectiveness Measures

The above variety in summarization methods for trend analysis is typical of a field that is still in its most creative phase. It is expected that as the combined evidence accumulates, some methods will begin to dominate over others. A second observation is that some summarizing methods are analogs of precision while others correspond to recall. For instance, the percentage of relevant pages retrieved over time (Cho et al., 1998) and the percentage of papers found as the percent of hyperlinks followed increases (Rennie and McCallum, 1999) are both estimates of recall. Similarly harvest rate (Chakrabarti et al., 1999; Chakrabarti et al., 2002b; Aggarwal et al., 2001), the average rank of retrieved pages (Menczer et al., 2001), and the average relevance of pages (Diligenti et al., 2000) are all estimates of precision (although the latter is within a sliding window).

Based upon our previous experience (Menczer and Belew, 2000; Menczer et al., 2001; Menczer, 2003; Pant et al., 2002; Menczer et al., 2004) and our study of the related literature we have selected for our framework a minimal set of measures that provides for a well rounded assessment of crawler effectiveness. In addition we propose performance/cost analysis as a way to gauge the effectiveness of the crawlers against their efficiency.

Table I. Evaluation scheme. S_c^t is the set of pages crawled by crawler c at time t. T_d is the target set and D_d is the vector representing the topic description, both at depth d. Finally σ is the cosine similarity function (Equation 1).

Relevance Assessments	Recall	Precision
Target Pages	$\lvert S_c^t \cap T_d \rvert / \lvert T_d \rvert$	$\lvert S_c^t \cap T_d \rvert / \lvert S_c^t \rvert$
Target Descriptions	$\sum_{p \in S_c^t} \sigma(p, D_d)$	$\sum_{p \in S_c^t} \sigma(p, D_d) / \lvert S_c^t \rvert$

Table I depicts our evaluation scheme. It consists of two sets of crawler effectiveness measures differentiated mainly by the source of evidence to assess relevance. The first set focuses only on the target pages that have been identified for the topic (row 1). For example the recall measure assesses the proportion of the target set retrieved at a given point of time during the crawl. The second set of measures (row 2) employs relevance assessments based on the lexical similarity between crawled pages (whether or not they are in the target set) and topic descriptions. Further details are given below. All four measures are dynamic in that they provide a temporal characterization of the crawl strategy. Dynamic plots offer a trajectory over time that displays the temporal behavior of the crawl. We suggest that these four measures of our framework are sufficient to provide a reasonably complete picture of crawler effectiveness — although they may not all be necessary or appropriate in the specific case of any given crawling application or experiment.

The rationale for effectiveness measures based on target pages is that we use the targets to approximate the actual, unknown set of all pages relevant with respect to the topic. Using Figure 5 as an illustration, we assume that the target set T is a random sample of the relevant set R. Therefore, for recall, $|R \cap S|/|R| \approx |T \cap S|/|T|$. This way it is possible to approximate actual recall using the existing classification by the directory editors.

The rationale for using effectiveness measures based on lexical similarity between crawled pages and target descriptions is that we want to assess a crawler's generalization power. To this end we need a source of evidence for topical relevance that is independent of the few keywords used by the crawling algorithms. Using the same topic representation for both crawling and evaluation would be akin to using the same data set to train and test a machine learning algorithm. The target descriptions are written by the editors and not accessible to the crawlers. But they are meant to describe the content of pages that are on topic,

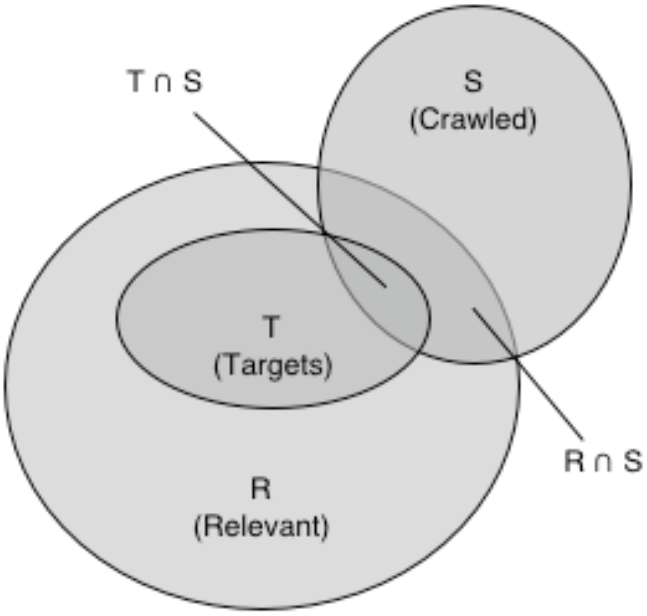

Figure 5. Relationship between target, relevant, and crawled page sets.

therefore they are a representation of the topic. If a crawler finds a page that is similar to the topic description, it is reasonable to assume that such a page may be topically relevant even it is not one of the target pages.

To assess page relevance using topic descriptions, the topic and retrieved pages must be represented using any reasonable, mutually compatible vector representation scheme. In our experiments topics and pages are represented by vectors of terms weighted by tf.idf (term frequency $\times$ inverse document frequency). Further details are provided in Section 5. Given topic and page vectors, D and p respectively, their similarity may be computed as their cosine, designated by $\sigma()$ in Table I:

$$\sigma(p, D) = \frac{\sum_{i \in p \cap D} p_i D_i}{\sqrt{(\sum_{i \in p} p_i^2)(\sum_{i \in D} D_i^2)}} \tag{1}$$

where p_i and D_i are the tf.idf weights of term i in page p and topic description D, respectively. Estimating page relevance through lexical similarity, we may approximate recall for the full crawl set by accumulating similarity over the crawled pages. The ideal crawler will achieve at each point of time the maximum possible similarity. Recall calculations would require division by the sum of similarity over all relevant pages. However, since this is a constant across crawlers and topics we may drop it from the calculations. For precision, the proportion of

retrieved pages that is relevant is estimated as the average similarity score of crawled pages.

In summary, our framework allows for a well rounded analysis with analogs of recall and precision performance measures using both a known target set of relevant pages as well as topic descriptions to assess the relevance of any crawled page. Finally, by plotting these measures over time, we get a dynamic characterization of performance.

3.3. Efficiency

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 processed text and links of fetched pages. Obviously the more complex the link selection algorithm, the greater the use of such resources. In order to allow for a fair comparison of crawling algorithms, our framework prescribes tracking the CPU time taken by each crawler for each page and each topic while ignoring the time taken by fetching, parsing and storing routines 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. The monitored CPU time will be used to compare the complexity of the crawling algorithms and gauge their effectiveness against their efficiency.

4. Characterization of Topics

The third dimension of our evaluation framework pertains to topic characteristics. In information retrieval research it is understood that query characteristics affect performance (Nelson, 1995; Saracevic and Kantor, 1998; Beaulieu et al., 2000; Mitra et al., 1998). In the classic 1988 study by Saracevic and Kantor (1998), query characteristics were explored within a larger context that included the study of users and search methods. Their questions were classified by expert judges regarding: domain (subject), clarity, specificity, complexity and presupposition. They found for example that the number of relevant documents retrieved was higher in questions of low clarity, low specificity, high complexity and many presuppositions. Beaulieu et al. (2000) correlated search outcomes with query characteristics examining aspects such as topic type. Mitra et al. (1998) explore the effect of query expansion strategies by differentiating queries based on their initial retrieval performance.

There is also active research on the types of queries users input to search engines (Spink et al., 2001; Jansen et al., 2000). For example

Spink et al. (2001) study over a million queries posed to the Excite search engine and find that the language of Web queries is distinctive in that a great many terms are unique.

Topic features are seldom explored in crawler research. An exception is when topic features are examined in order to elaborate on observed performance and to provide an explanation of results. For example, Chakrabarti et al. (1999) discuss a few of their twenty topics from Yahoo in detail in order to elaborate on their crawler mechanisms and to explore their notions of cooperative and competitive domains. Although Bharat and Henzinger (1998) do not differentiate between the 28 queries used to evaluate their topic distillation system, they do present results for the full topic set and for two special subsets: rare and popular topics as determined by the retrieval set size from AltaVista. Amento et al. (2000) experiment on a set of five topics that are somewhat homogeneous in that they are all representative of popular entertainment. Menczer and Belew (2000) test two crawlers (InfoSpiders and best-first) on topics from a limited Encyclopaedia Britannica (EB) corpus and analyze the dependence of performance on the depth of the topics within the EB subject hierarchy, where deeper topics are more specific.

In our general framework for crawler evaluation research, we seek to include consideration of topic characteristics that hold some potential for increasing our understanding of crawler performance. Thus our framework should allow one to look for significant correlations, positive or negative, between topic characteristics and performance. We begin our exploration by discussing the following four distinct characteristics:

Topic Popularity: Size of discourse set estimated by the number of pages containing topic keywords that are indexed by search engines;

Target Cohesiveness: Cliquishness of target pages in link space;

Target Authoritativeness: Average authority score of target pages among neighbor pages;

Seed-Target Similarity: Average similarity between seed pages and target descriptions.

To be accurate, the last characteristic is not really an inherent characteristic of a topic because it depends on the seeds, which are chosen by the user. Nevertheless we include seed-target similarity in our analysis because it is a part of the "operational" definition of a topic — crawlers cannot respond to topics unless provided with seed sets.

4.1. Popularity

Popularity indicates the level of interest in the topic. More popular topics will have larger numbers of interested individuals, related Web pages and discourse units. For instance "IBM computers" could be a more popular topic than "Web crawlers." We are interested in this property because it may be the case that crawler differences are accentuated when we pay attention to the popularity of topics. For example, some crawlers may perform more poorly on more popular topics if they are too reliant on lexical clues.

Topic popularity may be estimated by the size of its discourse set. One way to do this is to search for the topic keywords directly against a search engine and use the number of hits returned as an estimate. If multiple search engines are employed then the number of hits returned may be averaged. Recall that in our general framework we can associate a topic with the subtrees of the topic node at any `DEPTH` (up to `MAX_DEPTH`). Correspondingly, we can obtain popularity estimates dependent upon the value of `DEPTH` by using appropriate query representations of the topic when conducting the search. Thus we define *popularity* for a topic t at `DEPTH` d as:

$$P_d(t) = \frac{1}{|E|} \sum_{e \in E} \left[|H_e(K(t))| - \sum_{t' \in G_{d+1}(t)} |H_e(K(t'))| \right] \qquad (2)$$

where $K(t)$ is the keyword representation of topic t (i.e., the concatenation of node labels from the root of the directory to node t), $H_e(q)$ is the hit set returned by search engine e in response to query q, E is a set of trusted search engines, and $G_{d+1}(t)$ is the set of subnodes (subtopics) of t at depth $d+1$. Thus for $d = 0$ we look at the popularity of the topic in the most restrictive sense, excluding keywords of any subtopic. For d=`MAX_DEPTH`, we interpret the topic in a more inclusive sense, corresponding to the whole topic subtree. Note that the keywords in K are joined by AND syntax (all required) and thus P_d is by construction a non-decreasing function of d for any topic.

4.2. Cohesiveness

Topic cohesiveness estimates how closely knit the relevant pages are for a topic. The more cohesive a topic, the more interlinked its set of relevant pages. There are many ways to measure cohesiveness. Since we do not assume that a typical crawler has access to a search engine to find the inlinks of a page, for this measure we are particularly interested in the forward links that a crawler would observe locally during a

crawl. We start with the target pages of the topic as these are the ones assumed to be relevant. Cohesiveness is obtained by examining the neighborhood of the target pages. We use the forward links from all target pages and count the fraction of these that point back to the target set:

$$C_d(t) = \frac{\sum_{u \in T_d(t)} |O_u \cap T_d(t)|}{\sum_{u \in T_d(t)} |O_u|} \tag{3}$$

where O_u is the set of outlinks from page u. In essence, this measure estimates the likelihood of reaching another target page, given that the crawler has already located some target page. Note that since target sets are `DEPTH` sensitive, so is our topic cohesiveness metric.

Cohesiveness is a measure of the "cliquishness" of the target pages, and has been used in many contexts, for example to characterize the performance of a random-walk crawler (Menczer, 1997; Menczer, 2004) and to identify Web communities (Flake et al., 2000). We speculate that topics with high link cohesiveness could potentially make it easier for a crawler to stay within the vicinity of the relevant pages. This would be especially true for crawlers with localized search strategies.

4.3. Authoritativeness

The next topic characteristic metric in our framework pertains to authoritativeness. As proposed by Kleinberg (1999) a good authority is a page that has several good hubs pointing to it while a good hub is one that points to several good authorities. Kleinberg provides us with an algorithm that uses this recursive definition on a directed graph of Web pages to get authority and hub scores. We treat the target pages of a topic as a *root set* which is then expanded to get a *base set*. The expansion is done by including the pages corresponding to all the outlinks from the root set pages and the top I inlinks to the root set. Kleinberg's algorithm is then applied to the graph representation of the base set. Once the algorithm converges, we calculate the average authority score for the target URLs:

$$A_d(t) = \frac{1}{|T_d(t)|} \sum_{u \in T_d(t)} \Lambda(u, B(T_d(t))) \tag{4}$$

where $B(T)$ is the base set obtained from root set T and $\Lambda(u, B)$ is the convergence authority score for page u computed from base set B.

Since the authority scores are normalized, the average authority score $A_d(t)$, which we call *authoritativeness,* represents the concentration of authority in the target pages of topic t as inferred from their link based neighborhood. By taking target sets at different values of `DEPTH` d, we obtain depth sensitive estimates of topic authoritativeness.

4.4. Seed-Target Similarity

The last characteristic included in our framework is seed to target similarity. Here the point explored is that if the targets are lexically very similar to the seeds then it may be easier to reach the target pages. Thus we differentiate between topics on the basis of the average lexical similarity between the seed pages and the target descriptions:

$$L_d(t) = \frac{1}{|S(t)|} \sum_{u \in S(t)} \sigma(u, D_d(t)) \tag{5}$$

where $S(t)$ is the seed set for topic t.

Once again, seed page u and target description D_d may be any reasonable vector representation. Similarity σ is then defined as the cosine of the two vectors (see Equation 1). Typically, tf.idf weighted term vectors are used. Our specific implementation of weight representation is detailed in Section 5. As for the other characteristics, seed-target similarity is `DEPTH` sensitive. However, as mentioned above, this metric is also dependent on user-specified seeds, unlike the three preceding characteristics that are properties of the topic or targets not controllable by the user.

5. Case study

Our next goal is to demonstrate the application of the general evaluation framework presented in the previous sections, in an experiment comparing four off-the-shelf crawlers. In this case study we describe a specific implementation of the framework, i.e., a set of choices for parameter values and decisions related to the three dimensions of our evaluation: crawl task, performance measures, and topic characteristics.

The purpose of the case study is not to make claims about any particular task or crawling algorithm, but simply to give an example that illustrates how the general framework presented in this paper can be applied to evaluate and compare different crawlers in some well-defined crawling problem. A crawler designer or Web information retrieval practitioner will apply the framework specifically to the crawling techniques being considered, with a task suitable to the particular application of interest.

5.1. Crawl Task

The crawl task in our case study is motivated by applications in which Web pages are crawled while the user waits, for example to refine the

results of a search engine in order to find fresh hits which may not have been indexed by a search engine. An instance of such an application is the MySpiders applet[2] (Pant and Menczer, 2002). In such circumstance the number of pages crawled is severely limited, making it impossible to explore many promising links that are encountered during the crawl. To model this task we give crawlers a short lifespan of `N_PAGES` = 4000 pages. While this makes for a challenging and interesting problem, it is to be noted that crawlers designed for different applications, say building an index for a topical search engine, might be more appropriately evaluated crawling more pages.

We use the Open Directory hierarchy (`dmoz.org`) as our source for topics. Two key advantages of this choice are that (i) the ODP is maintained by a large number of volunteer editors and thus is not strongly biased toward commercial content, and (ii) it makes all of its data publicly and freely available through periodic RDF dumps.

We identified topics from this hierarchy at `TOPIC_LEVEL` = 3 and `MAX_DEPTH` = 2. By varying `DEPTH` from 0 to 2 we generated topic descriptions and target sets. Each topic node contributes to the topic description a concatenation of the text descriptions and anchor text for the target URLs, written by the ODP human editors (cf. Figure 2). Thus we have 3 sets of descriptions and 3 sets of target pages for each topic. The experiments described are differentiated by topic `DEPTH`.

Another reason for using a small `N_PAGES` in this case study is that given the limited CPU and bandwidth resources of any experimental setting, one must trade off crawl length versus number of topics. We used `N_TOPICS` = 100, a large enough number of topics to achieve statistically significant results, something that we believe should become the norm in Web information retrieval if we are to draw believable conclusions from crawling studies.

In addition, a set of keywords is defined for each topic. The keywords associated with a particular node are the words in the ODP hierarchy down to that node. The keywords are used to guide crawlers in their search for topical pages. For example the best links in a best-first algorithms are selected based on source page similarity to a topic representation build out of these keywords. Recall that our separation between the topic keywords passed to a crawler and the much richer topic descriptions used for evaluation is entirely intentional. First, keywords are more representative of the typical short queries that users employ to describe their information needs. Second, richer descriptions written independently by expert editors are key to assessing a crawler's focus and generalization effectiveness. Keywords corresponding to dif-

[2] `http://myspiders.informatics.indiana.edu`

Table II. A sample topic. For each DEPTH, only additional keywords, descriptions and targets are shown; the actual descriptions and target sets at each DEPTH d are inclusive of those for all DEPTH $< d$. Descriptions and target URLs are abridged for space limitations.

d	Keywords	Descriptions	Targets
0	Sports	ChairSports.com - Information and links on...	www.chairsports.com/
	Disabled	dizABLED: Wheelchair Stuntman Cartoons...	www.dizabled.com/
	Wheelchair	National Wheelchair Poolplayer Association...	www.nwpainc.com/
		Wheelchair Adventures - Discusses various...	www.afdl00104.pwp.blueyonder...
		Wheelchair Sports - A celebration of active...	lenmac.tripod.com/sports.html
		World Wheelchair Sports - Homepage of this...	www.efn.org/~wwscoach/
		Xtreme Medical Sports - Organization in...	www.xtrememedical.com/
1	Events	British Commonwealth Paraplegic Games- Brief...	www.internationalgames.net/br...
	Regional	Pan-American Wheelchair Games- Brief history...	www.internationalgames.net/pa...
		Stoke Mandeville Wheelchair Games - Brief...	www.internationalgames.net/st...
		The Wheelchair Bodybuilding Page - Lists...	www.angelfire.com/ky/thawes/
		Hamilton Wheelchair Relay Challenge...	www.hamilton-wheelchair-relay...
2	Australia	BlazeSports.com - A disabled sports program...	www.blazesports.com/
	Canada	Far West Wheelchair Sports- Events, results...	home.earthlink.net/~fwwaa/
	Hong Kong	Long Island Wheelchair Athletic Club (LIWAC)...	www.liwac.org/
	UK	Paralyzed Veterans Association of Florida...	www.pvaf.org/
	US	Sun Wheelers Sports- Non-profit organization...	www.geocities.com/sun_wheelers/
		BC Wheelchair Sports Association- Non-profit...	www.bcwheelchairsports.com/
		Canadian Wheelchair Sports Association...	www.cwsa.ca/
		Manitoba Wheelchair Sport Association- Sport...	www.sport.mb.ca/wheelchair/
		Ontario Wheelchair Sports Association Canada...	www.disabledsports.org/owsa.htm
		Wheelchair Sports Association Newfoundland...	www.netfx.ca/wsanl/
		i-Wheel Sports: NSW Wheelchair Sport...	www.nswwsa.org.au/
		New South Wales Wheelchair Sport - General...	www.isport.com.au/wheels/nswws/
		British Wheelchair Sports Foundation (BWSF)...	www.britishwheelchairsports.org/

ferent depths than the topic root node (DEPTH > 0) may also be used to compute topic popularity as described in Section 4.1. Table II provides as an example one of the 100 topics in our case study.

For seed selection we use the procedure described in Section 2.3. Backlinks are obtained via the Google Web API. Since the API has a limit of 1000 queries per day, we set N_QUERIES $= 1000$. The other parameters are DIST $= 2$ and N_SEEDS $= 10$. Thus at each iteration in the procedure we select n_sample $= 5$ backlinks. Barring any broken links, each of the 10 seed pages can lead to at least one target page at DEPTH $= 0$ within at most 2 links.

Data sets with the topics, keywords, targets, descriptions, and seeds used in this case study are available online.[3]

5.2. Evaluation Metrics

To evaluate the crawlers in our case study we follow closely the performance measures defined in Table I (Section 3.2).

When assessing relevance of the full crawl set against topic descriptions, both the target descriptions and the retrieved pages are pre-processed by removing common "stop words" and by a standard stemming algorithm (Porter, 1980). They are then represented by tf.idf vectors. Moreover, `DEPTH` dependent topic vectors are generated by concatenating the topic keywords and the topic descriptions down to the corresponding `DEPTH` d. Our idf calculations are also done with respect to the pool consisting of target descriptions down to `DEPTH` d in the topic subtree. We compute the tf.idf weight of term i in page p for topic t and depth d as follows:

$$p_{t,d}(i) = f(i,p) \cdot \left[1 + ln\left(\frac{|D_d(t)|}{|\{q \in D_d(t) : i \in q\}|}\right)\right] \tag{6}$$

where $f(i,p)$ is the frequency of i in p and $D_d(t)$ is the set of target descriptions for topic t and depth d. The tf.idf weights for topic vectors are computed analogously. For lexical similarity, the cosine formula in Equation 1 is used.

5.3. Topic Characteristics

In our case study we limit the analysis of topic characteristics to topic `DEPTH` $= 2$ only. We calculate topic popularity by searching each topic keywords against only the Google search engine, using the Google Web API. Searches are generated from the most inclusive interpretation of each topic, using just keywords at `DEPTH`=0. Topic cohesiveness has been fully specified in the discussion in Section 4.2. For topic authoritativeness, when generating the base set we use $I = 10$, i.e., we add to the base set the top 10 inlinks as retrieved by Google. This is due to the API's limitation of 10 results per query. We then apply Kleinberg's algorithm to this base set and calculate the authority score for each page in the target set as described in Section 4.3. For seed-target similarity, the pages (after stop word removal and stemming) are represented using tf.idf vectors (cf. Equation 6) and the cosine function defined in Equation 1 is used for similarity calculations.

[3] `http://www.informatics.indiana.edu/fil/IS/Framework/`

5.4. Crawling algorithms

The first use of the evaluation framework proposed in this paper should be to establish the "state of the art" in Web crawling algorithms. However, it is difficult to even choose candidate algorithms to evaluate because the many crawlers described in the literature are designed with different tasks in mind and implemented and tested with different methodologies. The performance assessments we find in the literature are mostly anecdotal in the absence of well-defined tasks, consistent evaluation measures, and sound statistical analyses.

As a starting point, for our case study we consider four crawlers based on various factors: (i) they are well known in the literature; (ii) they are well documented and therefore easy to reimplement; (iii) they represent different and well understood algorithms; (iv) they have been routinely used as baseline performers; or (v) they are novel versions of algorithms that have proved effective in our own prior research. Given that this is the first study formally comparing different crawlers based on a common evaluation framework, we believe its results are an important first step toward establishing the state of the art among topical crawlers. We hope that other researchers will challenge the crawlers outlined here and evaluate alternative algorithms that may produce provably better performance.

Figure 6 illustrates our architecture for crawling the Web according to the various algorithms. All crawlers are given the same topic keywords and seed URLs, and perform the basic procedure in Figure 7.

The comparison is made under the constraint of limited resources, i.e., we limit the memory available to each crawler by constraining the size of its internal buffer. This buffer is used by a crawler to temporarily store link data, typically a frontier of links that have not yet been explored. Each crawler is allowed to track a maximum of `MAX_BUFFER` links. If the buffer becomes full, the crawler must decide which links are to be substituted as the new ones are added. The value of `MAX_BUFFER` is set to 256 in our case study.

The crucial details that differentiate crawling algorithms are in the `process` function. The first crawler tested is a breadth-first crawler, which is the simplest strategy for crawling. It uses the frontier as a FIFO queue, crawling links in the order in which it encounters them. Najork and Wiener (2001) have shown that breadth-first crawlers effectively retrieve pages in PageRank order. The BreadthFirst crawler is used here mainly because it provides us with a baseline performance level that can help gauge the effectiveness of more sophisticated algorithms.

The next two crawlers are variations of best-first search (Cho et al., 1998; Hersovici et al., 1998). In its basic version (BFS1), given a frontier

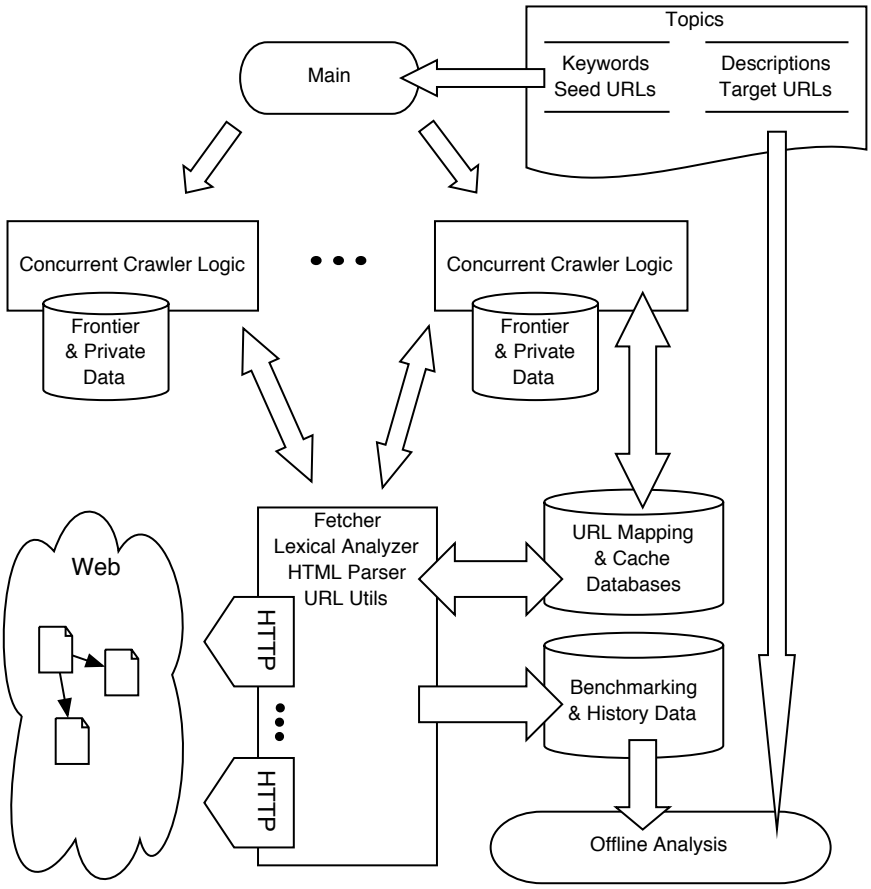

Figure 6. Architecture of our crawling system. Crawling algorithms run concurrently and are specified in modules that share common utilities (HTTP fetcher, HTML parser, URL processing, stopper, stemmer, lexical analysis, and benchmarking) and databases (cache and data collection). Each crawler module also maintains private data structures that are limited in size.

```
crawler (keywords, seeds, N_PAGES, MAX_BUFFER) {
  frontier = seeds;
  repeat N_PAGES times {
    link = process(frontier, keywords);
    new_links = visit(link);
    push(frontier, new_links);
    maintain(frontier, MAX_BUFFER);
  }
}
```

Figure 7. Pseudocode for the basic crawling procedure.

of links, the best link according to some estimation criterion is selected for crawling. BFSN is a class of crawling algorithms we introduced to generalize BFS1, in that at each iteration a batch of top N links to crawl are selected. Here we use BFS256, which has proved effective in our prior research (Pant et al., 2002; Menczer et al., 2004). Topic keywords are used to guide the crawl. Link selection occurs by computing the cosine similarity between the keyword vector and the source page vector, for each link. The N URLs with the best source page similarities are then selected for crawling. The worst links are eliminated from the frontier to make room for newly discovered, better ones when the frontier is full.

The last crawler tested is an implementation of InfoSpiders (IS) (Menczer, 1997; Menczer and Belew, 1998; Menczer and Belew, 2000; Menczer et al., 2004). In the basic IS algorithm, a population of agents crawls in parallel using adaptive keyword vectors and neural nets to decide which links to follow. An evolutionary algorithm uses a fitness measure based on similarity as a local selection criterion, and reinforcement learning to train the neural nets for predicting which links lead to the best pages based on their textual context in a source page. Agents that visit many pages similar to their internal keyword vectors get a chance to create offspring. An offspring inherits the keywords and neural net of the parent, modulo some mutations designed to internalize the features of the pages that led to the parent's success. The algorithm is completely distributed, with no interaction between distinct agents. Therefore the IS crawler can maximally exploit our concurrent architecture for efficiency.

The InfoSpiders implementation evaluated here includes a number of novel features designed to incorporate certain greedy aspects of BFSN that have made the latter more competitive in our prior experiments. Further details of IS and the other crawlers used in this case study are beyond the scope of this article. We refer the reader to a companion paper (Menczer et al., 2004) where these algorithms are described and analyzed at much greater depth.

5.5. Performance Analysis

Figures 8 and 9 show the performance analysis results for the four crawlers using our evaluation framework's effectiveness measures. The results are consistent across measures and with our prior experiments on these crawlers. In general we observe that BFS1 does well in the early stages of the crawl, but then pays a price for its greedy behavior (Pant et al., 2002). BFS256 eventually catches up, and in the case of target pages it outperforms the other crawlers. IS is outperformed by

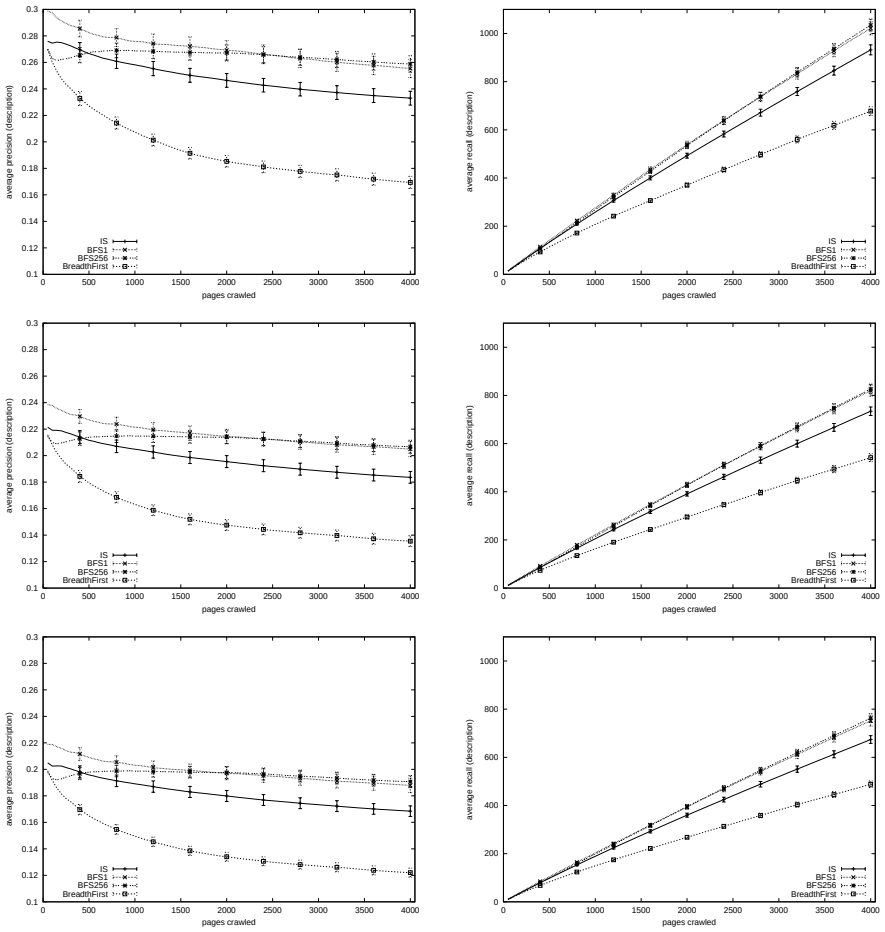

Figure 8. Dynamic plots of precision (left) and recall (right) versus number of crawled pages with relevance assessments based on target descriptions at `DEPTH` 0 (top), 1 (center), and 2 (bottom). Performance is averaged across topics and standard errors are also shown.

both BFS crawlers based on descriptions, while it almost matches the performance of BFS1 based on target pages. As expected BreadthFirst displays the worst performance and provides us with a baseline for all measures.

Precision and recall measures do provide complementary information in the evaluation. Precision captures a more dynamic and textured view of the behavior of the different crawling algorithms, especially in the early stages of the crawls. Recall provides for a clearer picture of the overall difference in the crawlers' asymptotic performance. Note that recall decreases with increasing `DEPTH`. This is because for most

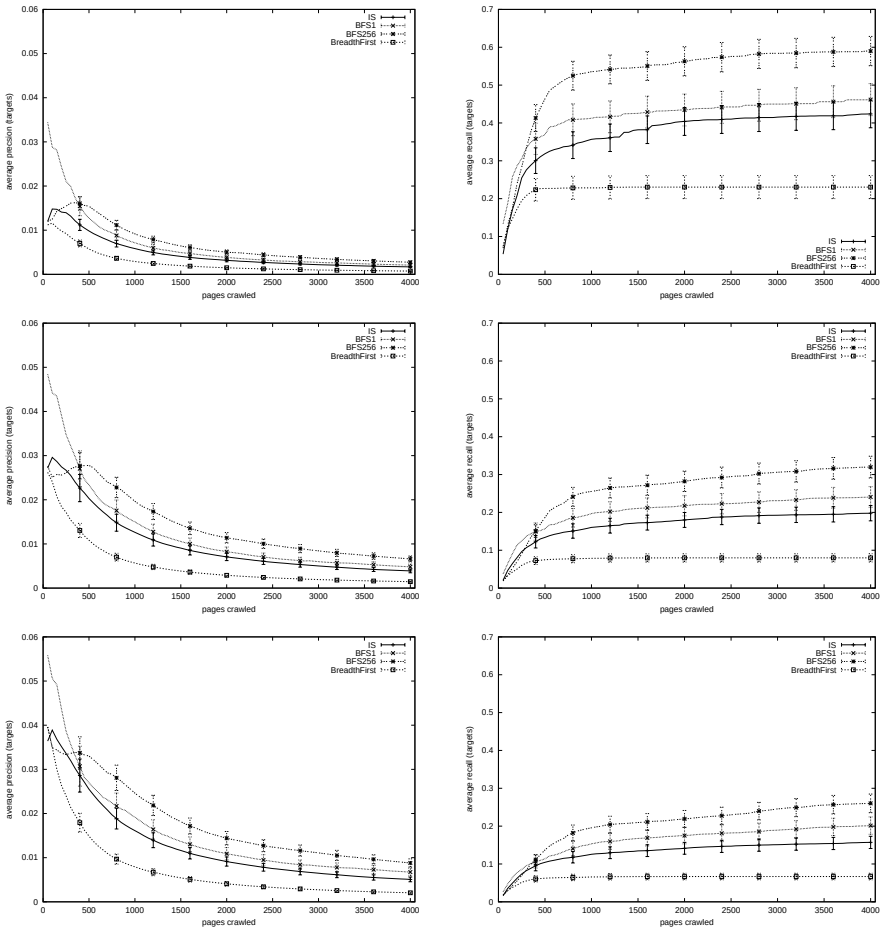

Figure 9. Dynamic plots of precision (left) and recall (right) versus number of crawled pages with relevance assessments based on target pages at `DEPTH` 0 (top), 1 (center), and 2 (bottom). Performance is averaged across topics and standard errors are also shown.

topics the number of target pages increases very quickly with `DEPTH`. So even as more targets are visited (cf. target precision in Figure 9), they represent a smaller fraction of the target set. It is also intuitive that the similarity between crawled pages and topic descriptions decreases as the latter become more inclusive and diverse; a relevant page may be similar to a few targets but dissimilar from many others.

To illustrate how performance is affected by the task, Figure 10 plots target recall for three of the crawlers in a more difficult task (`DIST` = 3) over a longer, 50,000-page crawl. (To remind the reader, the previous runs were done at `DIST` = 2 and over 4000-page crawls.) In

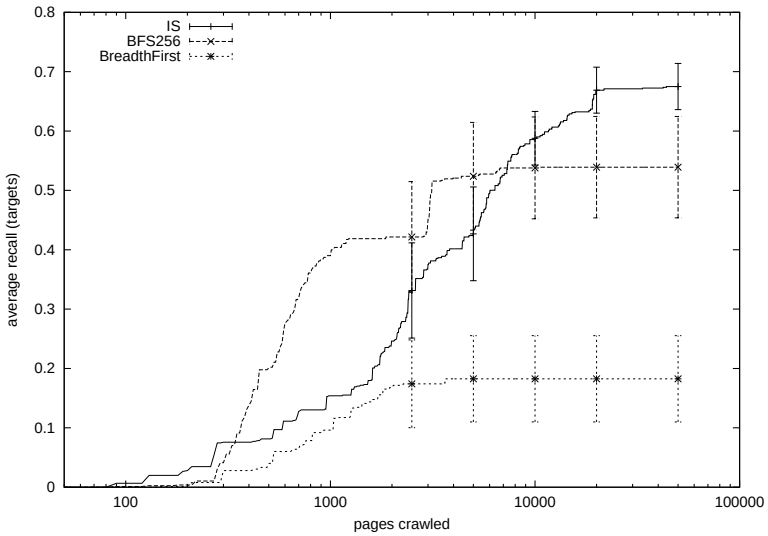

Figure 10. Average recall of target pages by BFS256, InfoSpiders and Breadth-First. The topics for this experiment are taken from ODP leaf categories, corresponding to maximum `TOPIC_LEVEL` and `DEPTH`=0. Other parameter values are `DIST`=3, `N_PAGES`=50,000, `MAX_BUFFER`=2,048, and `N_TOPICS`=10. Data from Menczer et al. (2004).

this experiment InfoSpiders eventually outperform the other crawlers, with the difference becoming significant after 20,000 pages (Menczer et al., 2004).

The above results draw a good picture of the different crawlers' effectiveness, but do not account for the computational complexity of the crawling algorithms. To gauge performance by the efficiency of the crawlers, Figure 11 shows the results of a performance/cost analysis based on our evaluation framework. Here we focus on recall measures, and on target descriptions and pages at `DEPTH` = 0. The results are quite interesting. Due to its efficiency BreadthFirst displays the best performance/cost ratio in the early stages of the crawl — if we need a few results really fast the simplest strategy may be the way to go. In the long run, IS achieves the highest performance/cost ratio thanks to its competitive performance and efficient use of concurrency. The BFS crawlers are penalized by less efficient algorithms, which (as implemented) require frequent sorting and synchronization operations (Menczer et al., 2004).

5.6. Topic Analysis

To analyze how crawler behavior is affected by different topics let us consider the correlation between performance and the various topic

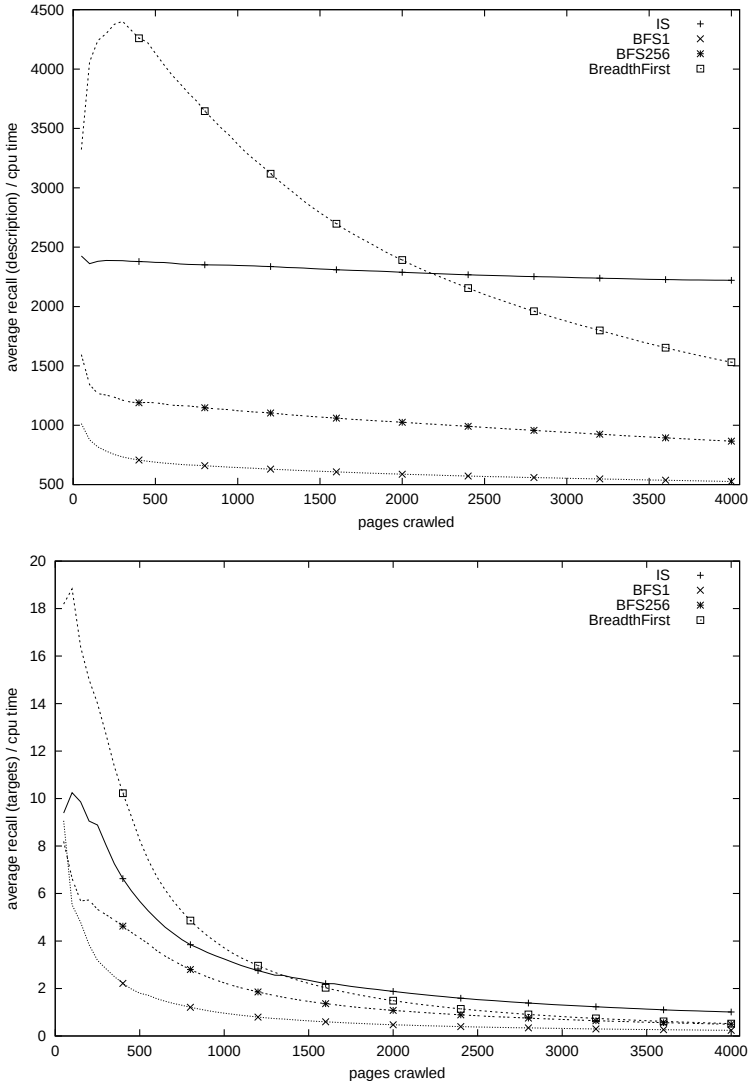

Figure 11. Dynamic plots of recall over relative CPU time for relevance assessments based on target descriptions (top) and target pages (bottom) at `DEPTH` = 0. CPU time must be normalized by its mean across crawlers to account for differences in the CPU speeds of machines used in our experiments (Menczer et al., 2004). Performance and CPU times are then averaged across topics before their ratio is computed.

Table III. Rank correlation coefficients between each crawler's recall after 4000 pages and the four topic characteristics for DEPTH=2. Recall is based either on target pages (left) or target descriptions (right). Values of ρ in bold indicate significant correlations at the 95% confidence level, based on a two-tailed Spearman rank correlation test (Conover, 1980). In these 14 cases we can refute the null hypothesis that there is no monotonic relationship between performance and topic characteristic.

	Target pages recall				Target description recall			
Crawler	C_2	A_2	P_2	L_2	C_2	A_2	P_2	L_2
IS	+.15	+.17	–.19	**+.54**	**+.41**	–.08	**+.20**	**+.37**
BFS1	+.03	–.01	–.18	**+.41**	**+.31**	–.06	+.07	**+.35**
BFS256	+.12	+.05	–.14	**+.53**	**+.32**	–.02	+.10	**+.41**
BreadthFirst	+.15	**+.27**	–.18	**+.31**	**+.36**	–.14	+.12	**+.28**

characteristics defined in Section 4. Here we need to pair a topic's characteristic with a crawler's performance; we use the cohesiveness, authoritativeness, popularity, and seed similarity measures at DEPTH=2 for the former, and the recall levels achieved by each crawler after 4000 pages for the latter. Since the distributions of all these measures are unknown, we need a distribution-free correlation measure and to this end we use Spearman's rank correlation coefficient ρ.

Table III shows the values of ρ for each crawler and topic characteristic, based on recall performance from target pages and target descriptions. Seed-target similarity is the topic characteristic that most significantly affects performance across crawlers. Higher seed-target similarity not only improves performance based on topic description, but also helps in reaching more predefined targets. The strong correlation may be indicative of the generally accepted principle that Web pages tend to point to lexically similar pages (Menczer, 2004). With that in mind, we also note that all of the topical crawlers (IS, BFS1 and BFS256) are more exploitative of seed-target similarity and hence show higher correlation than BreadthFirst.

While topic cohesiveness has no significant effect on target page recall, it does have a significant influence on description based performance. We interpret this observation by arguing that a cohesive topic may provide many paths to lexically similar pages even while identifying target pages may remain non trivial.

A topic's authoritativeness topic does not significantly influence any crawler other than BreadthFirst. Since the latter is not a topical crawler, it is able to improve its performance in reaching the targets simply because there are more paths leading to them — authoritative

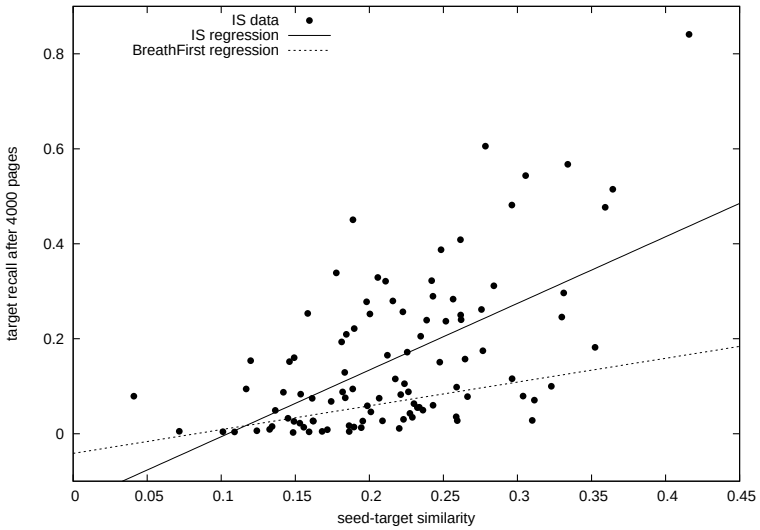

Figure 12. Scatter plot of target page recall for IS versus L_2. Each data point represents a 4000-page topical crawl. A linear regression is also shown for both IS and BreadthFirst.

targets are like attractors because they have many inlinks. This is consistent with observations that BreadthFirst crawlers effectively retrieve pages with high PageRank (Najork and Wiener, 2001).

Topic popularity seems to have contradicting effects on the two evaluation measures. Although only IS seems capable of exploiting this characteristic in a significant way, all crawlers tend to find more pages similar to the targets but fewer actual target pages for more popular topics. Our interpretation is that the large relevant set of a popular topic makes it easy to find many relevant pages, while it is hard to identify a relevant subset such as the target set.

As an illustration of the correlations in this data, Figure 12 shows a scatter plot of performance versus seed-target similarity for IS. For comparison, linear regressions are plotted for both IS and BreadthFirst. The plot makes it evident that IS tends to visit more relevant target pages when it starts from seeds that are lexically similar to the target descriptions.

6. Conclusions

In this paper we presented a general framework to evaluate topical crawlers. We identified a class of tasks that model crawling applica-

tions of different nature. By relying on Web directories, topics with the desired mix of specificity and inclusiveness can be easily identified.

The framework also specifies a procedure for defining crawling tasks of variable difficulty by selecting seed pages at appropriate distances from targets. The goal of such a formal and systematic characterization of crawl topics and tasks is to foster quantitative experiments that may allow researchers to better understand the differences between the many crawling applications found in the literature. To facilitate this endeavor, a script that selects topics from the Open Directory based on a number of parametric specifications, and generates files containing topic keywords, descriptions, target URLs at various depths, and seed URLs as illustrated in Section 2 has been made freely available under the terms of the Gnu General Public License.[4]

We introduced a set of performance measures to evaluate Web crawlers defined along several dimensions: precision versus recall, relevance criteria based on target pages versus human-compiled target descriptions, topic breadth, algorithmic efficiency, and dependence on diverse topic characteristics. Finally, we demonstrated the application of our framework in an experiment comparing four off-the-shelf crawlers. Our goal in the case study was to illustrate how the framework can be used to compare crawlers in a well-defined crawling problem.

6.1. Limitations

One important limitation of the approach underlying our general framework is its dependence on the availability of a hierarchical directory as a topic source. The Open Directory currently provides us with such a public resource, while other directories may be less open due to commercial concerns. Although it is possible to extend the framework to topic contexts that do not offer a hierarchical context, we do not address this aspect in this paper. A related limitation is that our framework makes the implicit assumption that these hierarchical structures effectively mirror the space of topics. The extent to which this assumption holds is unclear. Another limitation is that we have not considered user generated relevance judgments. Instead our framework considers external pages pointed to by the hierarchical directory pages as relevant. These manually identified pages are more appropriately considered topically relevant. Finally, we have studied topic characteristics such as popularity and cohesiveness as independent features. It remains to be seen if there are interactions between them.

[4] `http://www.informatics.indiana.edu/fil/IS/Framework/`

6.2. Implications

Given a particular crawling algorithm and a topic hierarchy one can use our framework to identify "good" algorithmic parameter settings for different topics and tasks. This would allow, for example, a vertical portal to use customized settings for crawling on each topic that it needs to index and update. In the absence of a topic hierarchy, appropriate parameter settings may be identified for a range of values corresponding to suggested topic characteristics such as popularity and cohesiveness.

The results of our case study clearly demonstrate that the proposed framework is effective at evaluating, comparing, differentiating, and interpreting the performance of diverse crawlers along all the studied dimensions. Topic analysis will give further insight into the behavior of crawling algorithms. Given a particular crawler, we may be able to predict its performance from the value of a topic characteristic, based on its sensitivity to that characteristic. For example we have shown that the IS crawler is most sensitive to the popularity of topics. Our results also show that all topical crawlers considered in the case study are more exploitative of seed-target similarity than the Breadth-First crawler. This is a validation of the hypothesis that topical crawlers effectively exploit topical locality (Davison, 2000; Menczer and Belew, 2000) on the Web. We also show that as the cohesiveness of topics increases all the crawlers seem to find more topically relevant pages. As a result of this finding, a crawling algorithm may be designed to look for cohesive subspaces within a topic since those subspaces can be expected to produce more relevant pages with less effort.

6.3. Further Research

While this is the most comprehensive treatment of topical crawler evaluation issues to date, it is only a first step. The Web information retrieval community now can use our evaluation framework to make objective comparative evaluations of alternative crawling algorithms and to advance the state of the art in a quantitative manner. It is to be emphasized that such advances will require appropriate statistical analyses (e.g., studies over many topics) in order to draw believable conclusions. Our framework allows one to develop and test additional topic characteristics. In future research we plan to explore characteristics such as recency and update frequency of topic target pages.

It is also desirable to experiment with the many parameters of our framework to achieve a better understanding of the factors that affect performance for different tasks and crawlers. Since the main emphasis of this paper is on presenting our evaluation framework, we did not exhaustively explore the role of parameters such as `DIST`, `TOPIC_LEVEL`,

and MAX_DEPTH. Given the general nature of our framework the space of possible experiments is quite large and it will take some time for the topical crawler community to identify the most useful task parameterizations. This endeavor is left for future research; the goal is to evaluate the many other crawlers in the literature and design better ones in support of a new generation of more scalable search tools.

Acknowledgements

Thanks to Alberto Segre, Dave Eichmann, Miguel Ruiz, Micah Wedemeyer and other colleagues for their support and contributions to this and our prior work. We are grateful to the Open Directory Project and its editors for their work and for making its data freely available, and to Google for their public Web API.

References

Aggarwal, C., F. Al-Garawi, and P. Yu: 2001, 'Intelligent Crawling on the World Wide Web with Arbitrary Predicates'. In: *Proc. 10th International World Wide Web Conference*. pp. 96–105.

Amento, B., L. Terveen, and W. Hill: 2000, 'Does "Authority" Mean Quality? Predicting Expert Quality Ratings of Web Documents'. In: *Proc. 23rd ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 296–303.

Beaulieu, M., H. Fowkes, and H. Joho: 2000, 'Sheffield Interactive Experiment at TREC-9'. In: *Proc. 9th Text Retrieval Conference (TREC-9)*.

Ben-Shaul, I. et al.: 1999a, 'Adding support for Dynamic and Focused Search with Fetuccino'. *Computer Networks* **31**(11–16), 1653–1665.

Ben-Shaul, I., M. Herscovici, M. Jacovi, Y. Maarek, D. Pelleg, M. Shtalhaim, V. Soroka, and S. Ur: 1999b, 'Adding support for Dynamic and Focused Search with Fetuccino'. *Computer Networks* **31**(11–16), 1653–1665.

Bharat, K. and M. Henzinger: 1998, 'Improved Algorithms for Topic Distillation in Hyperlinked Environments'. In: *Proc. 21st ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 104–111.

Brin, S. and L. Page: 1998, 'The Anatomy of a Large-Scale Hypertextual Web Search Engine'. *Computer Networks* **30**(1–7), 107–117.

Chakrabarti, S., B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg: 1998, 'Automatic resource compilation by analyzing hyperlink structure and associated text'. *Computer Networks* **30**(1–7), 65–74.

Chakrabarti, S., M. Joshi, K. Punera, and D. Pennock: 2002a, 'The Structure of Broad Topics on the Web'. In: D. Lassner, D. De Roure, and A. Iyengar (eds.): *Proc. 11th International World Wide Web Conference*. New York, NY, pp. 251–262, ACM Press.

Chakrabarti, S., K. Punera, and M. Subramanyam: 2002b, 'Accelerated Focused Crawling through Online Relevance Feedback'. In: D. Lassner, D. De Roure,

and A. Iyengar (eds.): *Proc. 11th International World Wide Web Conference*. New York, NY, pp. 148–159, ACM Press.

Chakrabarti, S., M. van den Berg, and B. Dom: 1999, 'Focused Crawling: A new approach to Topic-Specific Web Resource Discovery'. *Computer Networks* **31**(11–16), 1623–1640.

Cho, J., H. Garcia-Molina, and L. Page: 1998, 'Efficient Crawling Through URL Ordering'. *Computer Networks* **30**(1–7), 161–172.

Conover, W.: 1980, *Practical Nonparametric Statistics*, Chapt. 5, pp. 213–343. New York: Wiley.

Davison, B.: 2000, 'Topical locality in the Web'. In: *Proc. 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 272–279.

De Bra, P. and R. Post: 1994, 'Information retrieval in the World Wide Web: Making client-based searching feasible'. In: *Proc. 1st International World Wide Web Conference*.

Diligenti, M., F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori: 2000, 'Focused Crawling using Context Graphs'. In: *Proc. 26th International Conference on Very Large Databases (VLDB 2000)*. Cairo, Egypt, pp. 527–534.

Flake, G., S. Lawrence, and C. Giles: 2000, 'Efficient Identification of Web Communities'. In: *Proc. 6th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Boston, MA, pp. 150–160.

Henzinger, M., A. Heydon, M. Mitzenmacher, and M. Najork: 1999, 'Measuring search engine quality using random walks on the Web'. In: *Proc. 8th International World Wide Web Conference*. pp. 213–225.

Hersovici, M., M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur: 1998, 'The shark-search algorithm — An application: Tailored Web site mapping'. In: *Proc. 7th Intl. World-Wide Web Conference*.

Jansen, B., A. Spink, and T. Saracevic: 2000, 'Real life, real users and real needs: A study and analysis of users queries on the Web'. *Information Processing and Management* **36**(2), 207–227.

Kleinberg, J.: 1999, 'Authoritative sources in a hyperlinked environment'. *Journal of the ACM* **46**(5), 604–632.

Kumar, S., P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal: 2000, 'Stochastic models for the Web graph'. In: *Proc. 41st Annual IEEE Symposium on Foundations of Computer Science*. Silver Spring, MD, pp. 57–65, IEEE Computer Society Press.

Menczer, F.: 1997, 'ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery'. In: *Proc. 14th International Conference on Machine Learning*. pp. 227–235.

Menczer, F.: 2003, 'Complementing Search Engines with Online Web Mining Agents'. *Decision Support Systems* **35**(2), 195–212.

Menczer, F.: 2004, 'Lexical and Semantic Clustering by Web Links'. *Journal of the American Society for Information Science and Technology*. Forthcoming.

Menczer, F. and R. Belew: 1998, 'Adaptive Information Agents in Distributed Textual Environments'. In: *Proc. 2nd International Conference on Autonomous Agents*. Minneapolis, MN, pp. 157–164.

Menczer, F. and R. Belew: 2000, 'Adaptive Retrieval Agents: Internalizing Local Context and Scaling up to the Web'. *Machine Learning* **39**(2–3), 203–242.

Menczer, F., G. Pant, M. Ruiz, and P. Srinivasan: 2001, 'Evaluating Topic-Driven Web Crawlers'. In: D. H. Kraft, W. B. Croft, D. J. Harper, and J. Zobel (eds.):

Proc. 24th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval. New York, NY, pp. 241–249, ACM Press.

Menczer, F., G. Pant, and P. Srinivasan: 2004, 'Topical Web Crawlers: Evaluating Adaptive Algorithms'. *ACM Transactions on Internet Technology*. Forthcoming.

Mitra, M., A. Singhal, and C. Buckley: 1998, 'Improving Automatic Query Expansion'. In: *Proc. 21st ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 206–214.

Najork, M. and J. L. Wiener: 2001, 'Breadth-first search crawling yields high-quality pages'. In: *Proc. 10th International World Wide Web Conference*.

Nelson, M.: 1995, 'The effect of query characteristics on retrieval results in the TREC retrieval tests'. In: *Proc. Annual Conference of the Canadian Association for Information Science*.

Pant, G. and F. Menczer: 2002, 'MySpiders: Evolve your own intelligent Web crawlers'. *Autonomous Agents and Multi-Agent Systems* **5**(2), 221–229.

Pant, G., P. Srinivasan, and F. Menczer: 2002, 'Exploration versus Exploitation in Topic Driven Crawlers'. In: *Proc. WWW-02 Workshop on Web Dynamics*.

Pinkerton, B.: 1994, 'Finding What People Want: Experiences with the WebCrawler'. In: *Proc. 1st International World Wide Web Conference*.

Porter, M.: 1980, 'An algorithm for suffix stripping'. *Program* **14**(3), 130–137.

Rennie, J. and A. McCallum: 1999, 'Using reinforcement learning to spider the Web efficiently'. In: *Proc. 16th International Conf. on Machine Learning*. pp. 335–343, Morgan Kaufmann, San Francisco, CA.

Saracevic, T. and P. Kantor: 1998, 'A study of information seeking and retrieving. II. Users, questions, and effectiveness'. *Journal of the American Society for Information Science* **39**(3), 177–196.

Silva, I., B. Ribeiro-Neto, P. Calado, N. Ziviani, and E. Moura: 2000, '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*. pp. 96–103.

Spink, A., D. Wolfram, B. Jansen, and T. Saracevic: 2001, 'Searching the Web: The public and their queries'. *Journal of the American Society for Information Science* **52**(3), 226–234.

Srinivasan, P., J. Mitchell, O. Bodenreider, G. Pant, and F. Menczer: 2002, 'Web Crawling Agents for Retrieving Biomedical Information'. In: *Proc. Int. Workshop on Agents in Bioinformatics (NETTAB-02)*.

van Rijsbergen, C.: 1979, *Information Retrieval*. London: Butterworths. Second edition.