Using Metadata to Enhance a Web Information Gathering System

Neel Sundaresan
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120
neel@almaden.ibm.com

Jeonghee Yi
Computer Science, UCLA
405 Hilgard Av.
LA, CA 95 90095
jeonghee@cs.ucla.edu

Anital Huang
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120
ahuang@almaden.ibm.com

ABSTRACT

With the web at close to a billion pages and growing at an exponential rate, we are faced with the issue of rating pages in terms of quality and trust. In this situation, what other pages say about a web page can be as important as what the page says about itself. The cumulative knowledge of these types of recommendations (or the lack thereof) can be objective enough to help a user or robot program to decide whether or not to pursue a web document. In addition, these annotations or metadata can be used by a web robot program to derive summary information about web documents that are written in a language that the robot does not understand. We use this idea to drive a web information gathering system that forms the core of a topic-specific search engine.

In this paper, we describe how our system uses annotations about the hyperlinks contained in web pages to guide itself to crawl the web. It sifts through useful information related to a particular topic to eliminate the traversal of links that may not be of interest. Thus, the guided crawling system stays focused on the target topic. It builds a rich repository of link information that includes annotations. This repository is used to build quality metadata, which ultimately serves a search engine.

1. INTRODUCTION

The World Wide Web (web) today contains close to a billion pages and is growing at an exponential rate [12, 13]. As the number and complexity (in terms of the scripts, graphics, animations) of pages grow, to study and rate these pages based upon their content can become expensive and complex. As an alternative, it is possible to look at the pages that point to a page, and, to rate whether or not the page is of interest based on what other pages say about it. In this way, it is possible to learn from other people's experience.

Typical web crawlers crawl the web indiscriminately, paying little attention to the quality of search information. The goal of these crawlers is to get to as many pages as possible. Topic-directed crawlers have a different objective. Their goal is to get to all the pages related to their topic of interest *as fast as possible* without deviating to unrelated pages.

HITS[10] is a pioneering work that uses the link structure in hypertext documents to identify strongly connected components to discover high quality pages. HITS introduces the notion of authoritative pages and hub pages. Hub pages *point to* authority pages and authority pages are *pointed to by* hub pages. This system has been extended to build an automatic classifier [3], and a focused web crawler [4].

We start with the HITS premise and enhance it with descriptive information around links to identify the most appropriate metadata. We use a scheme that assigns weights to the edges in the link topology based upon the occurrences of certain topic words in the metadata to rate pages. We also use metadata to enhance abstracts, and, among other things, to decide on recrawl strategies.

The Organization of the Paper

The rest of the paper is organized as follows: Section 2 introduces the architecture of our web gathering system, called Grand Central Station. Section 3 introduces metadata in Web pages. Section 4 discusses how topic-specific gathering can be enhanced with link metadata. Section 5 reviews related work. Section 6 draws conclusions.

2. THE WEB GATHERER ENVIRONMENT

The experiments in this paper were conducted using the Grand Central Station (GCS) web gathering system [7]. This system was designed to be completely extensible and consists of an architecture that, at a high-level, includes 3 main parts: a *gatherer*, a *summarizer*, and a *topic expander*. Figure 1 shows the GCS architecture.

The *gatherer*is completely programmable in the sense that new protocols, data source types, and mechanisms for accessing and summarizing data can easily be added by writing Java classes with appropriate APIs. The gatherer uses an input specification file (based on XML, eXtensible Markup Language[2], syntax) to *plug in* specific protocol handlers, data source types, and summarizers, and, based on this specification, uses the Java reflection mechanism to load the appropriate Java classes. The gatherer, by itself, understands a dozen protocols (which include http, nntp, mail, jdbc etc.) and more than 50 file and mime types. Moreover, it can easily be enhanced or customized to target specific domains and has been customized in this way to build a number of domain specific search engines that include *jCentral* [23] (a search engine specific to Java-related information) and

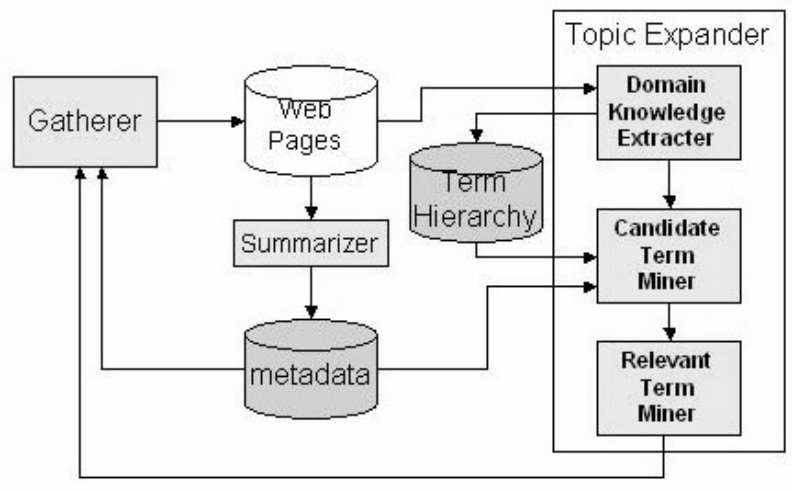

Figure 1: System Architecture. The *gatherer* visits web pages and the *summarizer* produces metadata on the links and the content of the pages. *Topic expander* identifies and extends relevant topic terms to the given topic pages on the basis of the metadata.

xCentral [24] (an XML-specific search engine).

The *summarizers* produce metadata from source documents in an RDF (Resource Description Framework) [6] format. RDF is used to build metadata graphs. The metadata summaries contain these graphs in the form of serialized XML. In addition to using RDF schema, the summary metadata uses an extensible schema architecture, provided by GCS, called *SumML*[17]. These XML metadata summaries go into a summary repository that is indexed to serve a front-end search engine. The advantage of producing RDF metadata is that it retains structure. As a result, end users can ask structural queries rather than simple boolean or structurally static queries as is the case with typical search engines like AltaVista [20], Infoseek [21], or Hotbot [22]. For instance, the RDF metadata summary for a Java program would contain structural information such as abstract information, class names, imported classes, inherited classes and interfaces, method names with signatures, and exceptions thrown. This structure enables a user to ask the search engine structured questions such as "show me all Java classes that implement the XML notation interface and are written by Kento Tamura".

The *topic expander* incrementally expands the topic definition as web crawling proceeds. Typically, a target topic consists of many subtopics. For example, the topic XML, according to the the Yahoo! [25] hierarchy (November 1999), consists of three subtopics: XSL (Extensible Stylesheet Language), SVG (Scalable Vector Graphics), and Software. In addition to its subtopics, the target topic also consists of related topics that share relevant properties but do not fall under the hierarchy. For example, topics relevant to but not subtopics of XML include SGML (Standard Generalized Markup Language), DTD (Document Type Definition), and XML Namespace. (Henceforth, we use the term *relevant topics* to refer to *both* sub-topics and related topics.) Exploiting hyperlink annotations, the topic expander learns of these relevant topics as the crawling proceeds. The details of the *topic expander* are out of the scope of this paper and presented in [19].

3. METADATA IN WEB DOCUMENTS

Hyperlink metadata, or simply *metadata*, is the information about a referenced document provided by the attributes of and the text around its hyperlink. As an analogy, consider the information about a research paper provided by its citation in another paper. In addition to the fact that the research paper was cited in the other paper, the citing paper offers the reader information about the cited paper. In web documents, citations are in the form of hyperlinks. In this paper, we concentrate on the hyerlink metadata in HTML documents (on which our experiments were run).

3.1 HTML Metadata

In HTML documents we find four kinds of hyperlinks:

- `anchor(<A>)` tags
- `Image(<IMG>)` tags
- `Map` and `Area` tags
- `Frame` and `iFrame` tags

Anchor tags are the most commonly used. They have several attributes associated with them. The main attributes include `name`, `title`, `alt`, `on-mouse-over`, and `href`. The Area tags have similar attributes. The IMG tag has other attributes such as `name`, `alt`, `src`, `dynsrc`, `lowsrc`, `onabort`, `onload`, and `onerror`. Some of these attributes are browser-specific but most are browser-independent. In addition to these attributes, the metadata information that we extract includes surrounding text. Our first goal is to identify the attributes that are most appropriate for the purpose of hyperlink metadata.

3.2 Metadata Extraction

In order to extract annotations from hyperlinks in an HTML document, we first convert the HTML document to a well-formed XML document. The browsers tend to be very forgiving and accept and display documents that are poorly

Metadata Type	Hyperlinks	Pages
`ALT` Tag	1,890 (0.9%)	281 (1.5%)
Anchor Text	147,745 (72%)	14320 (76%)
`HREF`	176,412 (85%)	16313 (87%)
`NAME`	5,487 (27%)	779 (4.1 %)
`ONMOUSEOVER`	9,383 (4.5 %)	1523 (8.1%)
Surrounding Text	49,138 (24%)	8424 (45%)
Title	885 (0.4%)	249 (1.3 %)

Table 1: This table lists various attributes associated with a hyperlink. The numbers in the column "Hyperlinks" measures the number of hyperlinks (and the percentage) which are referenced with a particular metadata type. The column "Pages" gives the number of pages which contains at least one particular metadata type.

formed HTML. We have an HTML to XML filter that converts the HTML documents to well-formed XML documents performing extensive error recovery in case of poorly formed HTML documents. Once we have a well-formed XML document, we check the document for elements with names corresponding to the hyperlink elements (`A` for anchor tags, `IMG` for image tags, and so on), and extract their attribute values. In order to identify the surrounding text, we identify XML elements of type PCDATA which are left and right siblings of these tags. For text contained inside an annotation tag, we look for PCDATA inside the HTML tags. Often, HTML pages contain a list of links to pages on a topic. The information about these topics is often provided at the parent level of this list. In order to identify in the XML document we pick the text data at the ancestral level where the closest sibling of a previous ancestor is a text node.

We studied a sample set of 20,000 HTML pages, which was collected by recursively visiting all hyperlinked pages from a given set of seed pages. We discovered over 206,000 hyperlink references from the sample set. Some of these hyperlinks pointed to pages within the sample set, while others pointed to pages outside the sample set.

Table 1 lists characteristics of these hyperlink metadata. It can be seen that, second to `HREF`, anchor text is the most common metadata type. In the results reported in this paper, we used anchor text because it was the most frequently occurring and the most reliable. Alternative schemes, like choosing weighted averages of different metadata type occurrences, however, may also be applied.

4. ENHANCING TOPIC-SPECIFIC INFORMATION GATHERING USING METADATA

This section demonstrates the utility of hyperlink metadata for topic-specific information gathering. We utilize metadata information i)for guiding crawlers to gather topic relevant pages efficiently, ii)for making recrawl decisions, and iii)for producing rich metadata summaries of web pages. This section begins with a presentation of the various algorithms for a topic-specific information gathering system followed by a review of their performance. It follows with an outline of metadata-based recrawl strategies, and concludes with a discussion of cumulative metadata in web page summaries.

4.1 Topic-Directed Crawling

Good pages for topic-specific information gatherers are the pages that pertain to the target topic. We have developed various algorithms for guiding topic-specific gatherers on the basis of hyperlink metadata. Due to the space limitation, we only briefly review the algorithms and present their performance on crawling good pages. For an in-depth description of the algorithms and detailed results from the crawling experiments, please refer to [18, 19].

4.1.1 HITS - Kleinberg's Algorithm

HITS [10] provides a weighting mechanism based on structural linking relationships among web pages. In general, a page is a good authority if many good hub pages point to it. Conversely, a page is a good hub if many good authority pages point to it. HITS computes hub and authority scores as shown in figure 2.

As proven by Kleinberg [10], the H and A vectors ultimately converge. In the crawling experiments, the system assigns hub and authority scores after iterating about n times (with n at times as low as 5). The crawler assigns priority according to the authority score of each page.

4.1.2 Simple Heuristics (SH)

SH first gives the highest crawl priority to those URLs that contain a predetermined set of relevant topic terms in their `HREF` metadata (for example, XML, DTD, or XSL). For those with same priority, the one with shortest depth gets highest priority. Using this algorithm, the crawler ultimately crawls all URLs descending from the seed set.

4.1.3 Relevance Weighting (RW)

With RW, the crawler selects and prioritizes pages on the basis of the relevance of a page. The relevance is measured on the basis of metadata. RW prunes pages that do meet a threshold value. $\Theta_{RW}(w)$, the RW relevance score of a page w, is computed as follows:

$$\Theta_{RW}(w) = \begin{cases} max(\Theta(t)_{t \in M(w)}) & \text{if } \max(\Theta(t)_{t \in M(w)}) \geq \text{c} \\ 0 & \text{otherwise} \end{cases}$$

where $\Theta(t)$ denotes the relevance of a term t to the target topic, which is discussed in detail in [18]. $M(w)$ is the hyperlink metadata of the in-link of w. c is a user defined threshold for relevance.

4.1.4 Relevance Weighting with Boosting (RWB)

The *RWB* technique deliberately increases the relevance score of some pages that fail to meet the relevance threshold. To be specific, for pages selected by random boosting, the algorithm recalculates relevance based on entire page content (rather than just the in-link metadata). Additionally, it boosts the scores of randomly selected pages that do not qualify even after this recalculation. $\Theta_{RWB}(W)$, the RWB relevance score of a page (w), is computed as follows:

$$\Theta_{RWB}(w) = \begin{cases} \Theta_{RW}(2) & \text{if } \Theta_{RW}(W) > c \\ & \text{or } R_1 < b \\ max(\Theta(t)_{r \in w}) & \text{if } R_1 \geq b \\ & \text{and } \max(\Theta(t)_{t \in w}) \geq c \\ R_2 & \text{if } R_1 \geq b \\ & \text{and } \max(\Theta(t)_{t \in w}) < c \end{cases}$$

1	Let P be a set of web pages.
	Let $E = \{(p_1, p_2) \mid$ Page p_1 has hyperlink to another page $p_2, p_1, p_2 \in P\}$.
2	For every page p in P, let $H(p)$ be its hub score and $A(p)$ its authority score.
3	Initialize $H(p)$ and $A(p)$ to 1 for all p in P.
4	While the vectors H and A have not converged:
5	- For all p in P, $A(p) = \sum_{(p',p)\in E} H(p')$
6	- For all p in P, $H(p) = \sum_{(p,p')\in E} A(w')$
7	- Normalize the H and A vectors.

Figure 2: Hypertext Induced Topic Search (HITS)

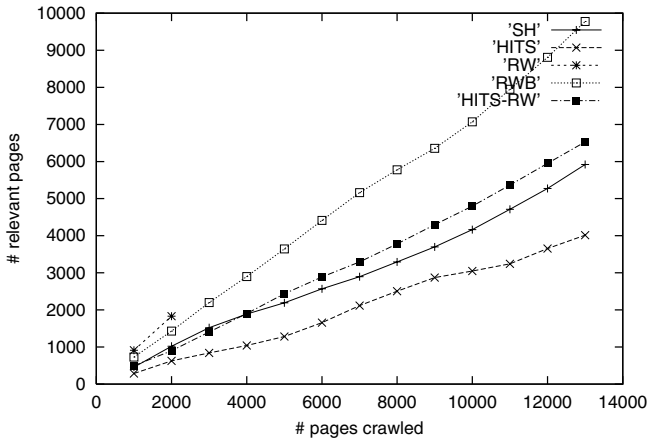

Figure 3: Quality of Crawling. The graph shows the total number of relevant pages discovered by each algorithm as the crawl progresses. RW achieves highest accuracy: 92% of pages crawled by RW are topic relevant. However, it stopped crawling after gathering about 3K pages due to the stagnation problem. RWB performs next best in accuracy, 75%. Moreover, it does not suffer from stagnation. By relevance edge weighting, the performance of HITS is improved from 31% (HITS) to 50% (HITS-RW). Yet the improvement is still marginal in comparison to the performance of RWB.

where R_1 and R_2 are random variables with uniform distribution, and b is a user-defined boosting factor.

4.1.5 HITS with RW (HITS-RW)

HITS algorithm is known to drift from the target topic [8, 1] because the algorithm is purely based on the link topology of web pages and does not take into account the relevance of each page. HITS-RW augments the algorithm by updating the adjusting the uniform link weights of HITS with the relevance of the hyperlink's metadata, With HITS-RW, an incoming link of a web page that has a higher relevance score for the topic lifts the *authority* score of the page more than the other links with lower scores. Ultimately, HITS-RW prioritizes web pages based on the relevance scores of the pages that point to them.

4.1.6 Topic-Directed Crawling Experiment

Figure 3 shows the effectiveness of various crawling techniques, HITS, SH, RW, RWB, and HITS-RW, measured by the total number of relevant pages identified by each technique as the crawl progresses.

RW gathers the highest quality web pages by avoiding unpromising hyperlinks, as expected. 92% of the web pages collected by RW were related to the target topic. In other words, decisions based on anchor-text metadata were accurate 92% of the time. The result was achieved without page lookups, i.e., downloading the pages. However, RW runs out of pages to crawl quickly because it over-prunes potential URLs to crawl.

Overall, RWB technique performs the best. RWB overcomes the over-pruning problem by visiting pages, with random probability, even if the metadata shows no promise. While this deliberate introduction of noise reduces the quality of the crawl, it allows the crawler to discover new clusters hence increasing the scope of relevant resources. Moreover, even with the noise, RWB does not extensively chase irrelevant links, in contrast to both SH and HITS, which do. RWB surpassed HITS-RW by a wide margin. We expected that HITS-RW would perform superbly with the combination of link and semantic weighting. Future work will investigate this observation more closely. As evidenced by the figure, the HITS performs poorly by neglecting page relevance.

These results support the use of hyperlink metadata for relevance measures. This behavior confirms that topic-directed crawling should take into account the page relevance and hyperlink metadata is an inexpensive estimation tool for measuring the relevance.

4.2 Recrawling

We use several different strategies to recrawl the web pages efficiently. The idea is to recrawl pages close enough to their time of change so that the search results are as up to date as possible. This means not recrawling pages that do not change and recrawling pages that change quickly, fast enough. In addition, given a choice between pages whose change frequency is unknown, we recrawl those that can result in pages with better content.

In the context of our discussion of metadata, we recrawl the hub pages more often because the chances of discovering new links from these pages is quite high. Since the hub pages are known to point to authority pages, if and when they change, they would probably add relevant links. Also, given a choice between two hub pages with the same relevance measure, we pick the one with the highest density of pointers to authority pages. Probabilistically, the chance of such a page adding a new authority page is higher than the one with a lower

density of pointers to authority pages.

4.3 Cumulative Metadata

The cumulative metadata that is produced for a web page contains link metadata information in addition to the summary of the data contained in the page. This includes all annotative metadata from this page about the pages that it points to and the annotative metadata by the pages that point to this page about it. Note that it is not possible at the time a page is crawled, to obtain exhaustive and up to date information on all the metadata that have been made about it. The reason is that some of the pages that point to the crawled page might not yet have been visited at the time when the page is crawled and summarized. However, when we recrawl the page, even if its content has not changed, if more pages that point to this page have been visited since the last crawl, the metadata for the page improves in quality in terms of its annotative metadata from other pages. Our link metadata repository gets richer as we continue to crawl, and the cumulative metadata for a page continues to be updated even without revisiting the page.

Example:
Suppose we have an HTML page at `http://www.xml.com/xpat` and two other pages `http://www.xmlauthority1.org` and `http://www.xmlauthority2.org` pointing to the first page. Suppose the first authority page has an anchor tag `<A HREF= "http://www.xml.com/xpat">Fast XML Parser</A>` and the second authority page has an anchor tag `<A HREF= "http://www.xml.com/xpat">Yet another SAX based XML Processor</A>`.

The summary for the page `http://www.xml.com/xpat` in XML-encoded RDF will be :

```
<rdf:RDF
 xmlns:rdf=http://www.w3.org/schemas/rdf-schema
 xmlns:gcs=
  "http://w3.almaden.ibm.com/gcs/summl-schema"/>
 <rdf:Description
  <!- attributes related to the web gatherer
      information ->
  resource="http://www.xml.com/xpat/">
 <rdf:Description>
  <parent-annotations>
   <!- annotations by the pages pointing to
       this page ->
   <rdf:Bag>
    <rdf:LI>
     <rdf:Description
      annotator="http://www.xmlauthority1.org"
      annotation="Fast XML Parser"/>
    </rdf:LI>
    <rdf:LI>
     <rdf:Description
      annotator="http://www.xmlauthority2.org"
      annotation=
        "Yet another SAX based XML Processor"/>
    </rdf:LI>
   </rdf:Bag>
  </parent-annotations>
  <annotations>
   <!- annotations by this page about the URLs
       it is pointing to -- >
    <rdf:Bag>
     <rdf:LI>
      <rdf:Description
       annotatee="http://www.xml.com/SAX processor"
       annotation="SAX information"/>
     </rdf:LI>
    <rdf:LI>
     <rdf:Description
      annotatee="http://www.xml.com/XML-press"
      annotation="XML Press Release"/>
    </rdf:LI>
   </rdf:Bag>
  </annotations>
  <!- other information from the web page being
      summarized ->
 </rdf:Description>
</rdf:RDF>
```

In summary, the `parent-metadata` property gives a list of metadata annotations by the pages that point to the summarized page. The `metadata` property gives the list of metadata annotations by the summarized page about the URLs that it points to. This information is indexed and fed to the search engine enhancing it in multiple ways. A good page that would not normally qualify under a search request might qualify because of its metadata. Alternatively, a page may be disqualified from the search result because it does not have strong enough annotative recommendations. The search engine also uses the metadata to enhance the abstracts of the search result entries.

The example shown here just shows plain text metadata. Our system defines a class hierarchy of metadata types. For instance, the metadata itself can be an XML fragment. The target search can be enhanced with support for RDF schemas [6] that understands the different metadata types.

5. RELATED WORK

HITS[10] introduces a novel mechanism to rate web documents in identifying good authority pages based on links. Subsequent work on classification [3], focused crawling [4], and web trawling for identifying micro-web communities [11] are based on the HITS. [9] stores important pages in a separate queue and visits them first on the basis of both content and link similarity measures. Their similarity measure is the occurrence of topic word in the content. Unlike our system, this system does not use metadata available about a page to avoid visiting the page.

Cora [14] defines domain-specific search engine for computer science papers. It uses machine learning techniques to improve the crawls. Chen *et. al.* [5] also refine the techniques of utilizing metadata for information retrieval. SPHNIX[15] has a classifier which annotates pages and links with various types of metadata to build web-site specific and customizable crawlers. ParaSite[16] is a system that exploits link information (not just hyperlinks) to build applications like finding individuals homepages, or expired or moved pages. Our system is novel in that it uses XML to build metadata structure for web pages. This metadata information is

augmented with weighted measures of authority pages and association mining techniques to improve the quality of the crawls.

6. CONCLUSIONS

In this paper we described our work on using metadata to build a topic-specific search engine. This metadata information includes what other pages say about this page. We use this information to direct our crawler to a particular topic of interest. The crawling strategy utilizing hyperlink metadata significantly improved the relevance of the gathered pages to the topic. By RW algorithm, the crawler gathered more than 92% of web pages that are relevant to the topic, but suffered from stagnation. With RWB algorithm, the crawler continuously crawled over 75% of relevant web pages without stagnation. It is worth reminding that the result was achieved in parallel to the minimum degree of unnecessary download of pages. This improved the crawling performance significantly.

Metadata for the crawled pages are described in XML-ised RDF graphs, and is included in the summary of the page. This is used to enhance search results and abstracts of search results. We have built a topic-specific search engine for XML based on the techniques described here.

7. REFERENCES

[1] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proc. of* 21^{st} *International ACM SIGIR Conference*, Melbourne, Australia, 1998.

[2] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. *Extensible Markup Language (XML) 1.0, W3C Recommendation*. World Wide Web Consortium, Feb. 1998. http://www.w3.org/TR/1998/REC-xml-19980210.

[3] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. of the* 7^{th} *International World Wide Web Conference*, Brisbane, Australia, Apr. 1998.

[4] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *Proc. of the* 8^{th} *International World Wide Web Conference*, Toronto, Canada, May 1999.

[5] H. Chen, Y. Chung, M. Ramsey, and C. Yang. A smart itsy bitsy spider for the web. *Journal of American Society of Information Science*, 49(7):604–618, 1998.

[6] R. G. Dan Brickley. *Resource Description Framework (RDF) Schema Specification, Proposed Recommendation*. World Wide Web Consortium, Mar. 1999. http://www.w3.org/TR/PR-rdf-schema.

[7] M. Eichstaedt, D. Ford, R. Kraft, Q. Lu, W. Niblack, and N. Sundaresan. Grand central station. Technical Report IBM Research Report, IBM Almaden Research Center, Aug. 1998.

[8] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *HyperText*, pages 225–234, Pittsburgh, PA, 1998.

[9] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. In *Proc. of the* 7^{th} *International World Wide Web Conference*, Brisbane, Australia, Apr. 1998. Also available from http://www-db.stanford.edu/pub/papers/efficient-crawling.ps.

[10] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of* 9^{th} *ACM-SIAM Symposium on Discrete Algorithms. Also appeared as IBM Research Report RJ 10076 (91892)*, May 1997.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proc. of the* 8^{th} *International World Wide Web Conference*, Toronto, Canada, May 1999.

[12] S. Lawrence and L. Giles. Searching the world wide web. *Science*, (280):98–100, Apr. 1998.

[13] S. Lawrence and L. Giles. Accessibility and distribution of information on the web. *Nature, 400*, pages 107–109, July 1999.

[14] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Building domain-specific search engines with machine learning techniques. In *AAAI Spring Symposium*, 1999.

[15] R. Miller and K. Bharat. Sphnix: A framework for creating personal, site-specific web crawlers. In *Proc. of the* 7^{th} *International World Wide Web Conference*, Brisbane, Australia, Apr. 1998.

[16] E. Spertus. Parasite: Mining structure information on the web. In *Proc. of the* 6^{th} *International World Wide Web Conference*, Santa Clara, California, Apr. 1997.

[17] N. Sundaresan and D. Ford. An architecture for summarizing the web. In *Proc. of the International Conference on Metadata*, Montreal, Canada, Aug. 1998.

[18] J. Yi, N. Sundaresan, and A. Huang. Automated construction of topic-specific web search engines with data mining techniques. Technical Report IBM Research Report, IBM Almaden Research Center, Apr. 2000.

[19] J. Yi, N. Sundaresan, and A. Huang. Metadata based information gathering. In *International database Engineering and Applications Symposium, forthcoming.*, Yokohama, Japan, Sept. 2000.

[20] Altavista. http://www.altavista.com/.

[21] Infoseek. http://www.infoseek.com/.

[22] Hotbot. http://www.hotbot.com/.

[23] jcentral. http://www.ibm.com/developer/java/.

[24] xcentral. http://www.ibm.com/developer/xml/.

[25] Yahoo! http://www.yahoo.com/.