On semi-automated Web taxonomy construction

Ravi Kumar* Prabhakar Raghavan† Sridhar Rajagopalan* Andrew Tomkins*

Abstract

The subject of this paper is the semi-automatic construction of taxonomies over the Web. We address the problem of discovering high-quality resources that belong in a particular node of a taxonomy. We show that minimal additional effort is required to provide relevance feedback in a hyperlinked environment, resulting in significant and consistent improvement in quality. Furthermore, this feedback is especially valuable for topics for which it is more difficult to find high-quality pages. Enroute, we describe novel algorithms for hyperlink relevance feedback.

Keywords: taxonomy construction, ontology, link relevance feedback

1 Overview

The Web has reached a point where even a large team of ontologists cannot hope to manually distill and maintain the best pages in a taxonomy of tens of thousands of topics. Yahoo! reportedly uses hundreds of ontologists to maintain its taxonomy; they classify (mostly submitted) URL's into Yahoo's tree. There are two main advantages to this approach. First, the taxonomy contents are generally *relevant*—humans judge far more accurately than computers that, for example, a page is about music production rather than music promotion. And second, ontologists provide valuable editorial annotation in the form of pithy one-line page summaries; such annotations are difficult to produce automatically.

On the other hand, relevance and quality differ for two reasons: (1) in a process driven largely by submissions, the pages that are listed are ones whose authors want to be listed on a major portal, which may not be the pages of highest quality; (2) with the continued growth of the Web and the small amount of ontologist surfing time available per node to augment submissions, purely manual approaches cannot find high-quality pages about a topic as effectively as a high-quality tool that makes use of implicit judgments in the form of hyperlinks [3, 4, 2, 11, 12].

We consider the problem of generating high-quality, relevant, links for topics in a taxonomy tree, which can then be presented to a human ontologist for vetting and annotation. Our system is designed to be used in a two-phase taxonomy construction and maintenance process: (1) The ontologist uses a rich query language to specify the query for a node, allowing the system to generate a high-quality set of links about his topic. (2) The ontologist can then edit and annotate the resulting set of links to create an appropriate externally-visible node about the topic. We consider only the first phase of this process. The system we describe should be seen as part of a tool used by a human ontologist to create substantially larger taxonomies with the same investment of human effort. Our system is *not* a replacement for the ontologist.

To determine the effectiveness of such systems, we study the time/quality tradeoff for increasingly sophisticated approaches to specifying a query for a topic. We consider three modes of query. (1) *Naive queries* contain one or two terms similar to those typically received by traditional Web search engines. (2) *Advanced queries* incorporate an "advanced search" syntax allowing phrases, and plus/minus modifiers in front of terms and phrases. (3) *Exemplary queries* incorporate link-based relevance feedback. We describe (Section 3) the construction of three copies of a 450-node benchmark taxonomy (one for each of our modes). We show the following results (Section 4). First, naive queries can be entered with surprising efficiency (earlier results have already benchmarked the effectiveness of these queries [4]). Second, understanding a topic well enough to benefit from an advanced keyword query requires significant additional effort, but does not provide significant additional quality. And third, using exemplary queries, link relevance feedback can be performed with effort similar to that required for constructing an appropriate advanced keyword query, but the resulting improvement in quality is significant. These results are even more dramatic for topics for which the Web does not contain a large set of high-quality resources. While our results apply directly to the problems of taxonomy construction and maintenance, they also have implications for Web search.

*IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120.

†Verity Inc, 892 Ross Dr., Sunnyvale, CA 94089. This work was done while the author was at the IBM Almaden Research Center.

Conventional relevance feedback as studied in the information retrieval community [13] is well known to improve search results [6, 9], at the cost of increased complexity in the interface. However, in the domain of taxonomy construction and maintenance on the Web, human resources are the dominant costs for large taxonomy maintainers, and our results should be viewed from that perspective. For many topics, little effort may suffice to find the majority of high-quality pages, while for other topics, multiple iterations may be necessary. We show that more advanced techniques including link relevance feedback allow the ontologist to populate the difficult remaining fraction of nodes successfully, at some small additional overhead to the overall construction process. While it is perhaps unreasonable to expect users in the mass search market to subscribe to this paradigm, we show that it is an efficient way for ontologists to build high-quality resource lists.

We benchmark this process in the context of the Clever resource-gathering system [4]. In doing so, we present algorithmic extensions to the Clever system that support the semi-automated construction of a large taxonomy. In particular, we extend the system to accept example pages of various kinds into the specification of a query. The ontologist first issues a naive textual query and then selects appropriate pages in the return set as examples of either good resources, good "resource lists" (see below for a definition of this concept), or stop sites. Henceforth, we will refer to this particular form of link relevance feedback as *exemplification*. We augment Clever to incorporate exemplification into its core graph-theoretic algorithm (Section 2). Notice that exemplification is a complementary form of relevance feedback, not a replacement for traditional methods. A complete system will, and should, use both techniques.

2 Clever system and algorithmic modifications

The Clever system builds on the HITS algorithm due to Kleinberg [8]. Following Kleinberg's original paper, a number of modifications have been studied [1, 3, 4, 5, 10]. For a full description of the Clever system, see [4]; here we give a brief overview and discuss subsequent algorithmic modifications pertinent to the present work. The thesis underlying HITS, and Clever, is that content creation on the Web results naturally in two kinds of valuable pages: so-called *hubs* and *authorities*. Good authorities for a topic are pages that are definitive sources of information on that topic (e.g., `www.cnn.com` for the topic of daily news); they are pointed to by the good hubs for the topic. Good hubs for a topic typically contain many links to good authorities.

Given a traditional text query, Clever begins by obtaining an *initial set* of around 200 pages from an inverted index, such as the Altavista text-search engine. It then expands the initial set to generate the *root set* by adding any page pointing to, or pointed to by, a page in the initial set; typically, the root set contains a few thousand pages. Consider the graph in which there is a node for each page of the root set, with a weighted directed edge from a node to another if the former has a hyperlink to the latter. The weight of the edge is a function of the relevance of the text surrounding the anchor, augmented by several filters to determine, for instance, whether the same author created both pages. Having created this graph, Clever then determines hubs and authorities by applying the basic HITS algorithm augmented with a number of heuristics to address issues of the Web corpus such as mirrored pages, shared domain names, and so on (cf. [8, 4]).

Building taxonomies: Background and issues. In a preliminary experiment, we used Clever to build out a taxonomy tree of about 600 topics with about 70 human-hours of effort. This trial was instructive in a number of ways: (1) On over a third of the topics, a "naive" query consisting basically of the topic title with essentially automatic augmentations would yield high quality results with a precision of over 80%. (2) A topic could often be pinpointed through example terms alone. Thus, the query `Swissair KLM Sabena` would produce a good list of European airline companies, because it would snare good hubs through this query and then proceed to distill other good authorities such as British Airways and Lufthansa). (3) On some topics, Clever with a naive topic query would yield a mixture of excellent resources and some contamination from "nearby" concepts. This led us to ask whether one could re-run Clever on these results, with human relevance feedback? What would the mechanism (and human cost) of such feedback be? Standard notions from information retrieval such as example terms and stopwords could be extended to the graph-theoretic domain: one could potentially exemplify Web pages as *example authorities* (say `www.att.com` for long-distance phone companies) or *example hubs* (say `artorg.com/leonardo.htm` when the topic is Leonardo da Vinci), or identify *stopsites* (say `www.microsoft.com` when the topic is residential double-glazed windows) that might otherwise co-opt a topic due to their greater Web presence. How should one implement and exploit this paradigm of combined keywords, example pages, and stopsites?

We describe first new modifications to Clever for dealing with example hubs, example authorities, and stopsites.

We then present TaxMan, the graphical interface tool we have developed for administering such taxonomies.

Enhancements to Clever. We now describe our enhancements to the root set generation phase of Clever. During link relevance feedback, an ontologist may present the algorithm with : (1) example hubs, (2) example authorities, and (3) stopsites. These types of pages impact the algorithm in two ways. First, they influence the root set, the particular subgraph of the Web deemed central to the topic. Second, they influence the edge weights connecting hyperlinked pages. The following describes these issues in more detail:

NODE STRUCTURE. Each example hub and example authority is added to the initial set. Likewise, each page an example hub points to, and each page that points to two or more example authorities, is also added to the initial set. This has the effect of drawing in new hubs and authorities that are similar to the examples. Finally, no stop sites are allowed to enter the set.

EDGE WEIGHTS. Intuitively, the edges that point to example authorities or originate at example hubs should weigh more. Additionally, if a page is cited in the *lexical neighborhood* of citations to example authorities, then that link should weigh more. Let $w(x, y)$ denote the weight of the edge from x to y in the graph. The following four heuristics are in addition to the basic edge-weighting schemes stated in [3, 4]: (1) If x is an example hub and x points to y, then $w(x, y)$ is increased; (2) If y is an example authority and x points to y then $w(x, y)$ is increased; (3) If y is an example authority and x points to both y and y' in the same lexical neighborhood, then $w(x, y')$ is increased; and (4) If y and z are example authorities, and x points to y' in the same lexical neighborhood with both y and z and the reference to y' is between the references to y and z then $w(x, y')$ is increased. [1]

TaxMan. TaxMan is a tool for building hierarchical taxonomies. It is a simple Java-based GUI to the underlying extended query language. It has facilities to create, delete, modify, and traverse nodes of a taxonomy. It has support for entering query terms and runing the Clever algorithm on a node. The user can also select a particular site and add it as an example hub or as an example authority or as both (called example site) or as a stopsite. Various parameters (like number of hubs and authorities to display, a limit on the maximum number of URL's fetched per example hub and example authority, etc.) can also be tuned using TaxMan.

3 Experiment description

Our experiment involved taxonomy construction using a team of four ontologists (the authors of this paper). [2]

Building the taxonomies. Our experiment involved the construction of four taxonomies: three drawn from predefined subtrees of Yahoo! (Government, Recreation & Sports, and Science) plus a fourth "personal" taxonomy consisting of nodes of personal interest to one of our ontologists. There were between 100 and 150 nodes in each of the first three taxonomies, and 70 in the personal taxonomy, for a total of 455 nodes. We built each taxonomy three times, as follows:

(1) First, we described each node of each taxonomy using a "naive" query consisting essentially of the topic title, with (occasionally) some simple alternatives. For instance, for the `United Nations` node the naive query was `"United Nations" U.N.`. The intent was to simulate a near-automatic process that gives a very quick first cut at describing a node.

(2) Next we rebuilt the taxonomy using an "advanced text" query. For example, a node about US Airforce bases could contain the query `"united states air force bases" "usaf bases" "usaf base" "united states air force base" "Scott Air Force Base" "Altus Air Force Base" "Barksdale Air Force Base" -navy -army`). The intent is to simulate a richer text query incorporating domain knowledge gained by inspecting the results of the naive query.

(3) Finally, we rebuilt the taxonomy using exemplification. For example, the `solar power` node contained four example authorities, an example hub, and no stop sites.[3] These example pages were selected from the output of

[1] The magnitude of the various increases in weight depends on a number of factors. Consider searching for long-distance phone companies. If Sprint and AT&T are example authorities for this node, and both occur in a single list of links, we have strong evidence that the other elements of the list may be relevant to the topic. However if the list contains only AT&T then we have only weak evidence that the list is about long-distance phone companies. The increase in weight of an edge is a super-linear function of the number of links to example authorities occurring the edge, and of the proximity of the edge to these links.

[2] Thus, we understood the innards of the algorithm and were not "typical" ontologists. However, in discussions with a number of professional ontologists it repeatedly emerges that the intuitive notions of good hubs, good authorities, and good query construction are all that is really needed to implement our methodology—and these are readily comprehensible even to those oblivious to the details of the algorithm. In subsequent experiments, we have found that users with some familiarity with taxonomy management can quickly be trained to use our tools as or more effectively than we do.

[3] The example authorities were the International solar energy homepage (`www.ises.org`), The American solar energy society

Clever running on the advanced text query. We feel that this rich form of description, combining text and example sites, represents a new mode of Web resource gathering that exploits the nature of content creation on the Web in the hub/authority view.

Our goal in designing these experiments was to benchmark each mode of taxonomy construction, monitoring: (1) wall clock time elapsed during the construction of the taxonomy; (2) quality of resources found by each; (3) level of exemplification; (4) investment in looking at results of text searches. Our system was configured to log all the actions of our ontologists as they used TaxMan. These logs yield, among other things, the wall clock time used in taxonomy construction, the sequence of mouse clicks, the number of results pages viewed, etc. Together these give a comprehensive picture of the human ontological effort used in constructing taxonomies in the various modes.

Evaluating quality: The user study. As noted earlier, evidence from previous work [3, 4, 1] suggests that the average quality of the nodes we construct are comparable to, and often better than those of manually-constructed taxonomies, even using text queries only. In the evaluation of our taxonomies, therefore, we did not measure their quality against such manually-constructed taxonomies. Rather, our emphasis here is on the relative qualities of our three modes of taxonomy construction. Similarly, [1] compare the relative results of eight variants on HITS.

We collected user statistics evaluating the pages as follows. We collected 50 users willing to help in the evaluation of our results, and decided *a priori* that each user could reasonably be expected to evaluate around 40 URL's. Therefore, we needed to spread these 2000 total URL evaluations carefully across the well over 50,000 URL's contained in our taxonomy. We adopted a random sampling approach as follows. First, we constructed the entire taxonomy in each of the three modes of operation. After all three versions of the taxonomy were constructed, we randomly sampled 200 nodes for evaluation, chosen uniformly from all nodes. Thus each user would evaluate 4 topic nodes on average; given the 40-URL limit on user patience, this suggests that each user can be expected to view 10 URL's per topic node.

We configured Clever to return 25 hubs and 25 authorities for each topic node in each of the three modes of taxonomy creation, for a total of 150 URL's. Since we wish to ask each user to evaluate a total of around 10, we sub-sampled as follows. For a particular ordered list of URL's, we refer to the "index" of a particular URL as its position in the list—the first URL has index one, and so forth. Consider a topic node N. We chose a "high-scoring" index uniformly from the indices between 1 and 3, and a "low-scoring" index uniformly from the indices between 4 and 25. We then extracted the two hub (resp. authority) pages at these two indices in the list of hubs (resp. authorities), from the taxonomy constructed using naive queries. This resulted in four URL's. We performed the same extraction for topic node N in the advanced text and example modes of creation as well, resulting in a total of 12 URL's. These samples contained some overlaps however; in all the mean number of distinct URL's extracted per node was about 10.2. From classical statistics, the score we compute is an unbiased estimator of the actual scores (cf. [7]).

We then asked each user to evaluate four topic nodes from our 200, chosen randomly without replacement. Each user was provided with a Web page containing links to four topics. Clicking on a particular topic brought up a form listing the approximately 10 sampled URL's from that topic, with a set of radio buttons next to each URL. The values of the radio buttons were "unranked", "bad", "fair", "good", "fantastic" and "unreachable." The "unranked" selection was checked initially for each URL. Clicking on a URL opened that URL in a separate window, allowing users to browse through URL's without losing access to the evaluation form. At the bottom of the form, a submit button logged the rankings. Of our 50 users, 41 completed some node of the survey in time, and of the 146 nodes evaluated, 139 had one or fewer unranked nodes per page, so we performed our evaluations on these nodes, representing 1437 page judgments. Due to an error in logging, we lost almost 200 of these judgments and were therefore only able to incorporate 1240.

4 Results and conclusions

First, a word on evaluation. Pages ranked "unranked" (presumably because a user simply forgot to rank this page) or "unreachable", were not considered in the ranking. All other pages were assigned scores as follows: "bad" = 0, "fair" = 1, "good" = 2, "fantastic" = 3. In some situations, however, it is also interesting to consider an analog of the information retrieval measure of precision, representing the number of retrieved documents that are "on topic." We therefore define pages ranked "good" or "fantastic" as being on topic, and when we refer to precision values we mean under this binarization of our scores. This is conservative, since a "fair" page is considered (for the purposes

(`www.sni.net/solar`), The solar cooking archive (`www.accessone.com/~sbcn`), and Solarex (`www.solarex.com`). The example hub was Solar energy links (`zebu.uoregon.edu/eesolar.html`).

of precision) to be irrelevant. We do not define "recall" on the Web, as ground truth sets do not exist and would not remain current under the continued exponential expansion of content.

Ontologist effort. We specified naive queries using a flat file of node names (note that our taxonomy tree is pre-existing and fixed; we do not consider issues of structure creation). The naive query could simply be typed next to each node with no browsing overhead, no delays waiting for cgi scripts to return, no use of the mouse, and perhaps one keyclick overhead to move from one topic to the next. The naive experiment can therefore be seen as a lower bound on the possible time to specify content for a node. For naive queries, we logged overall wall clock time and found that each node took between five to ten seconds to specify on average depending on the ontologist.

Our timing results for the advanced and examplary modes of creation are shown in Table 1. Unlike the naive queries, these results include all the overhead of using TaxMan over a slow network. We therefore created a small number of naive queries using TaxMan in order to estimate the per-node delay inherent to the UI, and found that simply navigating from node to node, waiting for screens to repaint, and entering a single piece of data without any extraneous browsing required 25 to 40 seconds depending on the ontologist. As the figure below shows, timings range from under two minutes to about four and a half minutes per node. The government taxonomy proved to be difficult to specify quickly, since it often required significant browsing through `.gov` sites to find appropriate keywords and pages for exemplification. As the table shows, providing link relevance feedback through exemplification is roughly equal in overhead to providing advanced keyword search syntax in this experiment.

Effectiveness of link relevance feedback. Having considered the amount of effort required on the part of the ontologist, we now report the improvement in quality as a result of this effort. Figure 1 shows the average score for the top 25 documents under each of the three modes of creation. We see that there is no significant difference between naive and advanced queries, but there is a significant improvement for exemplary queries. Each point in all our graphs represents the mean of at least 30 samples.

Dependence on Web presence of topics. We now examine the differences in the quality of pages discovered from one taxonomy to the next. Figure 1 shows for each possible page index (1–25) the average score of all pages at that location or higher. Thus, the results are "cumulative." In order to get an overall measure of the difficulty of the topics, we included results from all three modes of creation in this aggregate. We see that it is more difficult to find high-quality pages for the personal taxonomy than for any of the other taxonomies. Furthermore, as the topics of the personal taxonomy contain fewer high-quality pages, average score falls off towards the tail of the list faster than it does for the general-purpose taxonomies.

Table 1 shows the average values over the top 25 results, broken down by mode of creation as well as taxonomy, in both the average score and the precision metrics. In both metrics, the quality of results for the personal taxonomy is lower; we address this issue in more detail below. As the table shows, relevance feedback using exemplification never significantly impacts the overall quality of the system; but in case of the personal taxonomy, it helps dramatically. In the government taxonomy, exemplification does not improve the quality of results; this occurs because multiple rounds of exemplification are occasionally needed as the ontologist comes to understand the nature of the topic, but we felt that the difficulty of measuring these multiple rounds would reduce the clarity of the results. Thus ontologists performed only a single pass in each mode of construction.

Performance by mode of creation. An examination of the nodes shows that topics in the personal taxonomy tend to be narrower in focus. For instance, some of the nodes are `FOCS/STOC, SIGMOD, WWW, Collaborative Filtering, Latent Semantic Indexing, Phrase Extraction, Kerberos, Smartcards`. There are far fewer pages about, for instance, the FOCS/STOC (theory) conferences than about the sport of ice hockey. Interestingly, in this focused context we see the largest difference between modes: exemplification improved quality by approximately 33% over the purely textual approaches.

Conclusions. We draw two primary high-level conclusions from this work. The first conclusion, shown via our user study and the timing results of our instrumented taxonomy creation tool, is that an ontologist armed with the paradigm of iterative topic creation using increasingly sophisticated forms of query can create a high-quality taxonomy with a fairly quick turnaround time. The second high-level conclusion is that the well-known benefits of relevance feedback appear to hold in the domain of hyperlinked document search. As a tertiary conclusion, we show that, at least in the context of taxonomy creation, the traditional "advanced query" syntax used by search engines does not provide significantly better results than more naive queries. This might provide partial explanation for user dissatisfaction with "advanced search" functions in most search engines.

Taxonomy	Advanced	Exemplary	Naive		Advanced		Exemplary	
	secs.	secs.	Avg. Score	Prec.	Avg. Score	Prec.	Avg. Score	Prec.
Science	108.0	119.8	1.61	0.55	1.53	0.52	1.63	0.56
Recreation	192.4	239.6	1.64	0.61	1.68	0.64	1.70	0.63
Personal	157.5	214.0	1.03	0.30	0.91	0.31	1.41	0.48
Government	270.4	222.4	1.45	0.51	1.44	0.50	1.42	0.48

Table 1: Average construction time per node and average score, precision of top 25 hubs and authorities, by taxonomy.

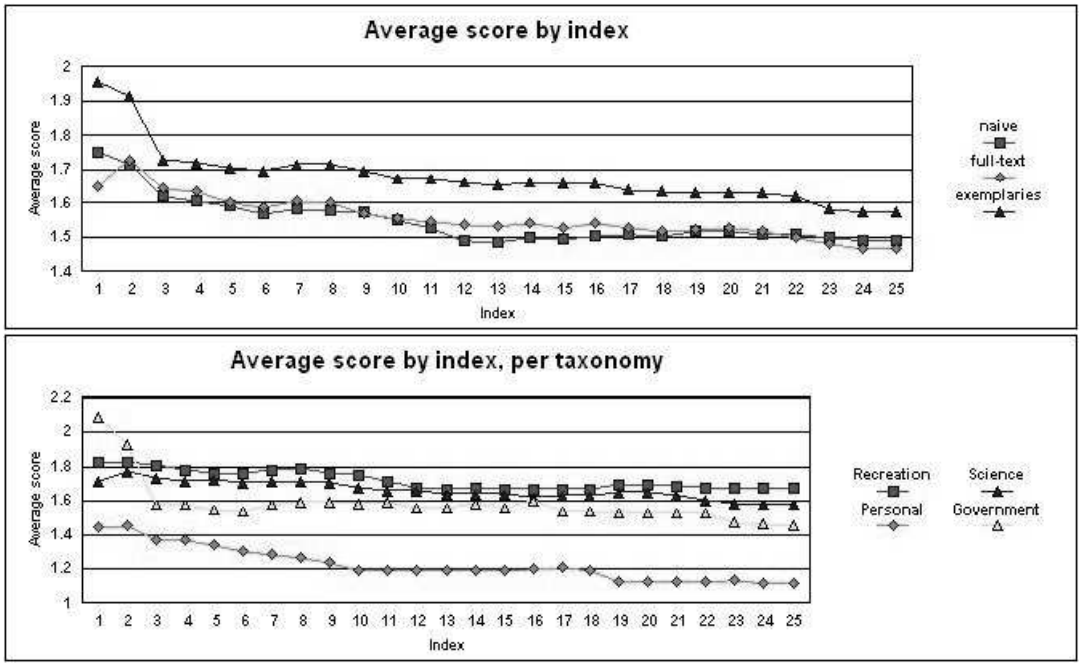

Figure 1: Average score by top 1 to 25 results, for naive, full-text, and exemplified queries and for each taxonomy.

References

[1] K. Bharat and M.R. Henzinger. Improved algorithms for topic distillation in hypertext environments, *Proc. 21st ACM SIGIR, 1998.*

[2] S. Brin and L. Page. The anatomy of a large scale hypertextual Web search engine, *Proc. 7th WWW*, 1998.

[3] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text, *Proc. 7th WWW*, 1998.

[4] S. Chakrabarti, B. Dom, Ravi Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Experiments in topic distillation. *Proc. ACM SIGIR Workshop on Hypertext Information Retrieval*, pages 13–21, 1998.

[5] J. Dean and M. Henzinger. Finding related pages on the Web, *Proc. 8th WWW*, 1999.

[6] E. Efthimiadis. *Interactive Query expansion and Relevance Feedback for Document Retrieval Systems*. Ph. D. Thesis, City University, London, UK, 1992.

[7] W. Feller. *An Introduction to Probability Theory and its Applications*. John-Wiley, 1968.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1998.

[9] J. Koenemann. Supporting interactive information retrieval through relevance feedback, *Proc. ACM SIGCHI*, 1996.

[10] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Proc. 9th WWW*, 2000.

[11] J. Pitkow and P. Pirolli. Life, death, and lawfulness on the electronic frontier. *Proc. ACM SIGCHI*, 1997.

[12] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the Web. *Proc. ACM SIGCHI*, 1996.

[13] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *JASIS*, 41(4):288–297, 1990.

[14] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.