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

* 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.

1 Overview

The Web has grown to a point where even a large team of ontologists cannot hope to manually distill (and maintain) the best pages on every one of tens of thousands of topics in a taxonomy. 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: (1) the pages classified into a node of the taxonomy are generally *relevant* — humans can do a far more accurate job than computers at judging, for example, that a page is about music production rather than music promotion; and (2) the editorial annotation provided by Yahoo's ontologists — pithy one-line summaries of the pages they list — is valuable to the user and difficult to accomplish using automatic methods.

On the other hand, relevance is not the same as quality, for the following 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 human 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. We employ an enhancement of the Clever [4] system as an underlying engine to generate pages given an ontologist-provided query, and we implement mechanisms for specifying queries that are a powerful augmentation to traditional keyword search. The 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 topic of a node, allowing the system to generate a high-quality set of links about his topic. We study this phase of the overall process in detail in this paper. (2) The ontologist can then edit and annotate the resulting set of links to create an appropriate externally-visible node about the topic. Hereafter, we do not consider this phase of the process. The system we describe can therefore be seen as a powerful tool used by a human ontologist, allowing substantially larger taxonomies to be created with the same investment of human effort; our system is *not* a replacement for the ontologist.

Our goal is to determine the time/quality tradeoff for increasingly sophisticated approaches to specifying a query for a topic. We consider three forms 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.

In the case of search systems, it is well-known that more sophisticated query interfaces can yield better search results. Conventional relevance feedback is very popular in the information retrieval community [13] and lot of work has been done to support interactive information retrieval through relevance feedback [6, 9]. On the other hand as the interface becomes more complex, it also becomes harder for a user to use it effectively. The consequence is a well-known quality–time tradeoff: as the time and effort spent specifying a query increases, so does the quality of search results. However, from the perspective of taxonomy construction and maintenance on the Web, in which the costs of human resources are the dominant costs for large taxonomy maintainers, our results should be viewed in a different light. Resource gathering is a multi-stage effort wherein an ontologist wishes to iteratively improve the quality of the resources corresponding to a topic. Thus, the ontologist engages a tool in an interactive process which successively improves the quality of the resource pages found as more time is invested by the ontologist into refining the quality of the search. 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. In doing so, we also describe new algorithmic extensions to the Clever system to support the semi-automated construction of a large taxonomy. More precisely, we extend the system to accept example pages of various kinds into the specification of a query. Specification of example pages is a natural form of link relevance feedback in a hyperlinked environment. 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 stopsites. 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 link 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]. The most recent of these [4] describes the basic Clever system; here we only discuss some 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 text query, in the first phase, Clever uses it to obtain up to 200 pages, called the *initial set*, from an inverted index (such as the Altavista text-search engine). Any page pointing to, or pointed to by, any of these 200 pages, is then added to the initial set to get the *root set*; typically, it contains a few thousand pages. Consider the graph in which there is a node for each page of this root set, with a 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 (1) the relevance of the text surrounding the anchor to the query and (2) whether the link is nepotistic or non-nepotistic. The second phase in Clever is to apply the basic HITS algorithm to this weighted graph. The Clever system also employs a number of different heuristics to specifically address the Web corpus, for instance, mirrored pages, or Web sites that contain multiple pages linked together (cf. [8, 4]).

Building taxonomies: Background and issues. The starting point for our investigations was the realization that (with some further modifications) the Clever engine could, with a small level of human effort, be used to build a taxonomy of topics. In a preliminary experiment, we were able to build 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 (possibly augmented by straightforward, essentially automatic query expansion — typically the expansion of acronyms) would yield very 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 for this topic through this query and then proceed to distill other good authorities for this topic, 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. Could one 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 alternatively 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 first phase of the Clever system for dealing with example pages. An interesting aspect of our algorithm is that it only makes use of the linkage neighborhood of an example page. The content of the page itself does not play a role. Potentially, a more sophisticated version would augment our method with more conventional relevance feedback studied in the IR literature [13]. Here we study link-based feedback methods because: (1) it is instructive to study the use of links alone; (2) they appear to serve our purposes quite well, as evinced by our results.

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. The example hubs and authorities and stopsites change the initial set and consequently the root set. An *example hub* and any page pointed to by an example hub are added to the initial set. An *example authority* is added to the root set and any page that points to at least two example authorities is added to the initial set. Finally *stop sites* are eliminated from both initial and root sets. The intuition behind these rules is the following. We believe the pages pointed-to by an example to be high quality candidate authorities. Therefore, pages that point to these pages have a better than average chance of being good hubs. So we add these candidate hubs to the root set by adding the example hub *and* its out-neighbors to the initial set. Similarly for example authorities: pages that point to two or more example authorities are high-quality candidate hubs, so we place them in the initial set so any candidate authorities they point to will be added to the root set.

EDGE WEIGHTS. Intuitively, the edges that point to example authorities or edges originating at example hubs should weigh more. Additionally, if a page is cited in the *lexical neighborhood* of 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.

The magnitude of the various increases in weight described above 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.

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 describing the query terms and run 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. Many of these parameters can be inherited in the hierarchy.

3 Experiment description

Our experiment involved taxonomy construction using a team of four ontologists (the authors of this paper). Perhaps the most notable consequence of this is that we understood the innards of the algorithm and were thus 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.

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 node `Government : International Organizations : United Nations` 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, describing each node using an "advanced text" query. Such a query could consist of descriptive terms as well as example terms (e.g., for the node `Government : U S Government : Military : AirForce : Bases`, the advanced text query could be `"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, possibly using some domain knowledge that the ontologist may have about the topic at hand; this is not something that can be automated. Such domain knowledge may have been obtained, for instance, by inspecting the results of the naive query.

(3) Finally, we rebuilt the taxonomy, giving Clever one or more example hubs and authorities (e.g., for the node Science : Energy : Solar Power, the example authorities were International solar energy homepage (www.ises.org), The American solar energy society (www.sni.net/solar), The solar cooking archive (www.accessone.com/∼sbcn), and Solarex (www.solarex.com) and the example hub was Solar energy links (zebu.uoregon.edu/eesolar.html). These example pages were selected from the output of Clever running on the advanced text query. This is the richest form of description in our experiment — a combination of text and example sites. The exemplification was done using the TaxMan interface described above. We believe this combination of text with example hubs and authorities represents a new mode of Web resource gathering, one that exploits the nature of content creation on the Web in the hub/authority view.

Some remarks about our experimental setup at this point. First, it is likely that in practice, a human ontologist would begin with what we consider to be an advanced text query. The naive query is of interest, nevertheless, since it represents what is possible using a minimal amount of human effort, and possibly even automated with a sophisticated wordnet. In fact, on a number of nodes, our naive query would yield very focused results; as we quantify in our report of the results below, exemplification (either through additional search terms, or through example hub and authority pages) frequently did not help the quality very much in these cases. This brings us to a second point: in the process of supplying advanced search terms (in the advanced text query phase) or example pages (for the third phase), the ontologist spends time viewing pages from the results of previous runs; we should account for this time.

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 quantitative picture of the human ontological effort used in constructing taxonomies in the various modes, where the time is spent, and what fraction of the time the ontologists spend significant time browsing and exemplifying. It remains to describe how we analyzed the quality of the taxonomies that were built using these 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. See also [1], who compare the relative results of eight variants they propose 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.

Clever returns 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 to mean 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 $h(N)$ uniformly from the indices between 1 and 3, and a "low-scoring" index $l(N)$ uniformly from the indices between 4 and 25. We then extracted the two hub (resp. authority) pages at indices $h(N)$ and $l(N)$ 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. Note that unlike some previous user studies on HITS and its descendants [1, 3, 4], we do not tell our users whether a particular URL was generated as a good hub or as a good authority; we will return to this point later. The evaluation methodology worked as follows. Each user was provided with a Web page containing links to four topics (the four topic nodes mentioned above), and was instructed to judge each URL in the topic as "bad", "fair", "good" or "fantastic" (or "unreachable" if the link is broken) based on how useful the page would be in learning about the topic. 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 we show that minimal additional effort is required to provide relevance feedback in a hyperlinked environment, via exemplification, resulting in significant and consistent improvement in quality. Conversely, however, we show that the additional effort needed to provide meaningful advanced keyword query syntax does not generate such an improvement in quality. Next, we establish that the benefits of link relevance feedback are especially significant for topics for which it is more difficult to find high-quality pages.

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. When we refer to scores in the following, we mean these values. 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. Note that this is conservative, since a "fair" page is considered (for the purposes of precision) to be irrelevant. Notice that there is no good way to define the notion of "recall" on the Web, as ground truth sets do not exist and would not remain current under the continued exponential expansion of content — hence, we restrict our attention to precision.

Effectiveness of link relevance feedback. In this section we consider the amount of ontological effort required for the three modes of taxonomy creation, and the resulting levels of page quality. We begin with results for 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 example 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; in fact, in the government taxonomy, it was actually less expensive to perform exemplification.

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 4 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 these curves 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 for each taxonomy. Figure 4 shows for each possible page location 1 through 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

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.

included results from all three modes of creation in this aggregate. Once again, each point in these curves represents the mean of at least 30 samples. 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 performance 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.

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.

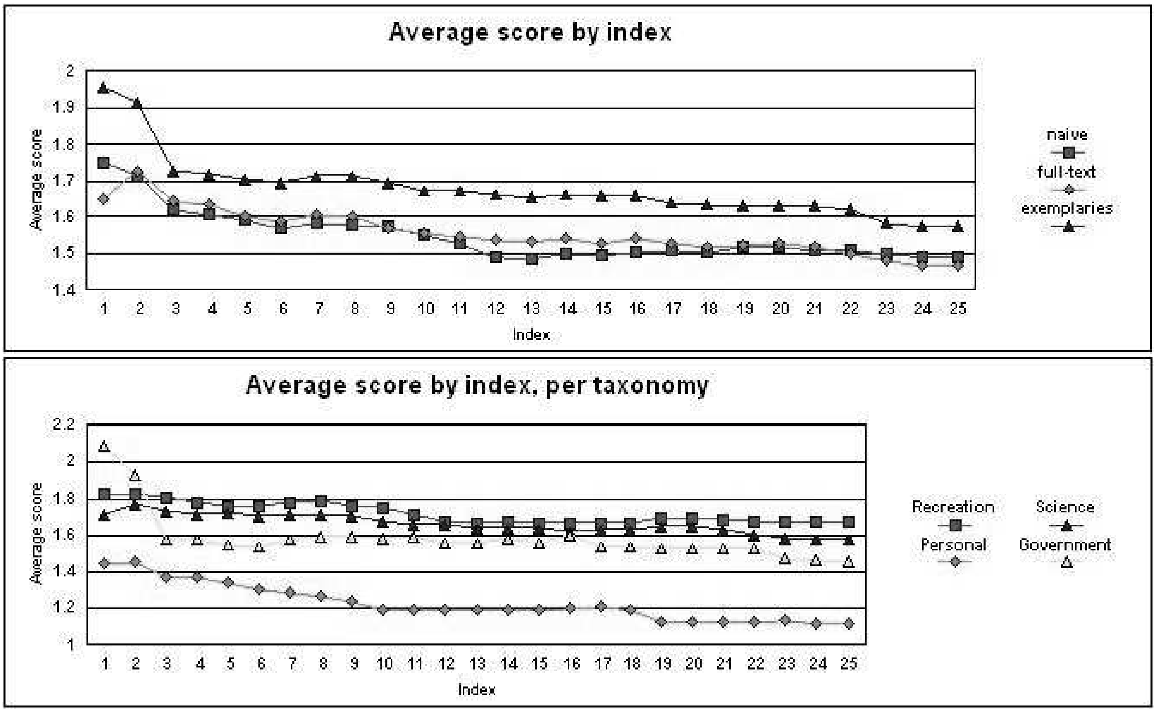

Figure 1: Average score by top 1 to 25 results, for naive, full-text, and exemplified queries and for each taxonomy.

[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.