RJ 10200 (95074) November 15, 2000
Computer Science

Research Report

HUMAN EFFORT IN SEMI-AUTOMATED TAXONOMY CONSTRUCTION

Ravi Kumar
Prabhakar Raghavan
Sridhar Rajagopalan
Andrew Tomkins

IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

RJ 10200 (95074) November 15, 2000
Computer Science

HUMAN EFFORT IN SEMI-AUTOMATED TAXONOMY CONSTRUCTION

Ravi Kumar
Prabhakar Raghavan
Sridhar Rajagopalan
Andrew Tomkins

IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Abstract

We describe the semi-automated construction of a 450-node topic taxonomy using the Clever system [CDK+98]. We focus on three paradigms using which an ontologist can specify topics to the system, requiring a different orders of magnitude of human effort (ranging from a few seconds to a few minutes). The most sophisticated of these is driven by a combination of keywords and exemplified pages of various types. This appears to be an emerging mode for information gathering that raises many interesting questions. We benchmark the quality of the resources gathered by the Clever search engine in each paradigm, and the human effort required, and derive several insights into the level and effectiveness of human effort in taxonomy construction.

Human effort in semi-automated taxonomy construction

Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins
IBM Almaden Research Center K53, 650 Harry Road, San Jose, CA 95120, USA.

Abstract

We describe the semi-automated construction of a 450-node topic taxonomy using the Clever system [CDK+98]. We focus on three paradigms using which an ontologist can specify topics to the system, requiring a different orders of magnitude of human effort (ranging from a few seconds to a few minutes). The most sophisticated of these is driven by a combination of keywords and exemplified pages of various types. This appears to be an emerging mode for information gathering that raises many interesting questions. We benchmark the quality of the resources gathered by the Clever search engine in each paradigm, and the human effort required, and derive several insights into the level and effectiveness of human effort in taxonomy construction.

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 between fifty and a hundred 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: it is difficult to maintain a high level of authority and quality for the pages listed in such a manually-constructed taxonomy, 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 explosive 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 automatic resource compiler [CDK+98].

We consider the problem of automatically generating high-quality, relevant, links for topics in a taxonomy tree, which can then be presented to a human ontologist. We employ the Clever system as an underlying engine to generate pages given an ontologist-provided query, and we suggest mechanisms for specifying such queries that are a powerful augmentation to traditional keyword search. We imagine this system being used by human ontologists in two ways. First, the ontologist can use 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 element of the problem in detail in this paper. Second, the ontologist can then edit and annotate the resulting set of links to create an appropriate externally-visible node about the topic. 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.

Given this "semi-automatic" mode of operation, we wish to allow the ontologist to generate high-quality lists of resources about a topic efficiently, using a rich and powerful query language. In the case of search systems, it is known that more sophisticated query interfaces can yield better search results [R79] . 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. We view this from a different angle: 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 the tool in an interactive process which, in the ideal case, successively improves on 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. While it is 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. In fact, as our experiments will show, our process frequently yields high-quality resources with minimal effort and, in these cases, more sophisticated iterations do not yield better results (and frequently make things worse).

We benchmark this process in the context of the Clever resource-gathering system. To this end, we describe (a) an interface for resource gathering, called TaxMan (for Taxonomy Manager); and (b) a simple methodology by which an ontologist can create resource lists following an iterative refinement process such as the one suggested above. In doing this, 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 Clever system to take both a standard keyword based query as well as example pages of various kinds into account when computing its response.

The specification of example pages is a form of relevance feedback. The ontologist first issues a naive textual query and then picks some of the pages returned as examples of good resources, or good lists of resources --- a process we refer to as "exemplification." The Clever system then recomputes its response taking this additional information into account. We propose a simple and effective new technique for folding these example pages into the computation of relevance. Unlike relevance feedback for traditional corpora, our proposed technique makes use of both the linkage structure and the content of the example pages. We do not propose this as an alternative to the more classical methods for relevance feedback, but as a complementary method. A complete system will, and should, use both techniques.

Early in the course of developing this system, we built a 600-node taxonomy using roughly 70 man-hours of human effort. While this is somewhat striking, it became clear that it was necessary to better understand such issues as: (1) to what extent is this process replicable? (2) how should ontologists describe a topic effectively to an automatic resource compiler? (3) how does the quality depend on the kinds of topics being covered?

We report on a detailed investigation into the human effort involved in using such a system, in the course of building a new benchmark 450-node taxonomy (including a "Personal taxonomy"). We have instrumented the process to extract various measures of ontologist productivity. Our results show that it is possible to gather high-quality resources to populate such taxonomies at widely-varying levels of topic detail, with human effort ranging from a few seconds to a few minutes per topic, depending on the topic. We detail the quality of the resources discovered, through a user-study. We correlate the quality of the search results to the effort of the ontologist in creating the query. As a by-product, we derive several insights on the process by which ontologists can effectively discover resources for taxonomy construction. Finally, we present and evaluate a system for populating topic nodes based on keywords in addition to various types of example pages, and we show that the incorporation of example pages significantly improves the Clever resource compilation tool.

Organization of this paper

In Section 2 we begin with a brief review of Kleinberg's HITS algorithm [K98], and descendants [BH98,CDG+98,CDK+98]. Next, we motivate several new algorithmic modifications to Clever driven by the needs of taxonomy construction. We conclude Section 2 with a description of TaxMan (Taxonomy Manager), a GUI tool, which provides an intuitive interface to the Clever search engine on one hand and to the topic tree on the other. We begin Section 3 by motivating and describing three modes in which an ontologist can describe a topic to Clever. We next describe the construction of three copies of our test 450-node taxonomy (one for each of our modes). We also describe the techniques used in measuring both ontologist effort on one hand and result quality on the other. In Section 4 we give the results of the experiments, and the conclusions we derive from them.

2. Clever system and algorithmic modifications

Review of HITS and descendant work

The Clever system builds on the HITS algorithm due to Kleinberg [K98]. Following Kleinberg's original paper, a number of modifications have been studied [BH98,CDG+98,CDK+98]. The most recent of these [CDK+98] describes the basic Clever system; here we only discuss some subsequent algorithmic modifications pertinent to the present work.

We begin with a brief description of the basic Clever system. 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. Kleinberg breaks the apparent circularity in these definitions using an iterative calculation. This may be viewed as computing the principal eigenvectors of two related matrices derived from a subgraph of the web defined by the query topic. The computation results in a ranking of web pages by their hub and authority scores for the topic.

Given a text query, Clever uses it to obtain up to 200 pages, called the *initial set,* from an

inverted index (such as the text-search engine Altavista). Any page pointing to, or pointed to by, any of these 200 pages, is then added to the initial set. Call this enhanced pool 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 (a) the relevance of the text surrounding the anchor to the query and (b) whether the link is nepotistic or non-nepotistic. Each node maintains a hub score and and an authority score, both initialized to 1 and subsequently updated by the following iterative calculations: (1) the hub score of each node v is replaced by the weighted sums of the authority scores of all nodes pointed to by v; (2) the authority score of each node v is replaced by the sum of the weighted hub scores of all nodes pointing to v. Under a suitable rescaling, these updates converge to a steady state, under which each node is assigned a hub score and an authority score.

The algorithm 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. The reader is referred to [K98] and [CDK+98] for details. Several papers [BH98,CDG+98,CDK+98] have proposed broad classes of modifications with a view towards further improving the search results: (1) by weighting each link depending on the presence of the search terms in the pages at the two ends of the link; (2) various devices for preventing a single site from gaining a disproportionate influence or representation in the results; (3) assigning scores to a (contiguous) subset of a page --- rather than the entire page --- for better focus when the page contains other topics besides the query topic; (4) heuristics to increase the number of distinct authorities covered by the hubs, and for dealing with mirrored hubs.

The work leading to Clever [CDG+98,CDK+98] compared the system's search capabilities with commercially available search systems. Bharat and Henzinger [BH98], on the other hand, present a comprehensive study of the effects of using various subsets of the modifications sketched above, studying the relative quality of these variants.

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 man-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). A conventional text index could not be expected to exploit such exemplification.
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 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 traditional relevance feedback studied in the IR literature. 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 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

Recall that the graph is constructed from the root set, which is in turn constructed from an initial set. The example hubs and authorities and stopsites change the initial set and consequently the root set. In the case of an example hub, the following rule is applied:

> **Example hubs:** The example hub is added to the root set. Any page pointed to by an example hub is added to the initial set. Thus, if x is an example hub, and x points to y, then y is added to the initial set.

In the case of an example authority, the rule is more sophisticated. Pages that point to more than one example authority are added to the initial set. This is reflected in the following rule:

> **Example authorities:** An example authority is added to the root set. Any page that points to at least two example authorities is added to the initial set. Thus, if x points to y and z both of which are example authorities, then x is added to the initial set.

Finally stop sites are eliminated from both initial and root sets.

> **Stopsites:** If s is a stopsite, then s is deleted from both the initial set as well as the

root set.

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 [CDG+98][CDK+98].

Example hubs: If *x* is an example hub and *x* points to *y*, then *w(x,y)* is increased.
Example authorities: If *y* is an example authority and *x* points to *y* then *w(x,y)* is increased.
Lexical neighborhood 1: 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.
Lexical neighborhood 2: 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 notion of "lexical neighborhood" may be implementation-specific and we can use some heuristics (for instance, to determine whether two links occur on opposite sides of a heading or separator) to decide whether links are in the same neighborhood.

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.

Once the graph is constructed as above, the original Clever algorithm is run on this graph to obtain a list of hubs and authorities. Therefore all the modifications described above do not impact the convergence characteristics of Clever.

TaxMan

TaxMan is a tool for building taxonomies. It is a simple web-based GUI (built using Perl and CGI) to the underlying extended query language. Taxman can create hierarchical taxonomies. 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.

The following figure shows a screen snapshot of TaxMan in action. The node is Science : Earth Sciences : Oceanography. The fields containing the query terms, example hubs, example authorities, and stop sites can be seen on the left frame. The options add/repl/sub control the manner in which these values are inherited --- whether they replaced the existing values of this parameter (ie, specifying a new query), supplement them (ie, specifying an additional stopsite without overwriting other stopsites), or reduce them (ie, turning off certain stopsites in a node about search engines). The fields global/local determine whether inheritance occurs. The right frame contains the hubs and authorities output by Clever. In this frame, prefixing each hub/authority is a selection (seen as H A E S) using which each link can be enrolled as an example Hub, example Authority, Example site, or Stop site. For example, the first authority in the main frame is the NOAA homepage, which has been added to the "example_auths" field in the left frame by clicking the "A" button next to the link. The top frame contains some administrative controls (like logging facility) and controls for modifying the structure of the taxonomy, navigating between taxonomies, and using automatically-generated maintenance queues.

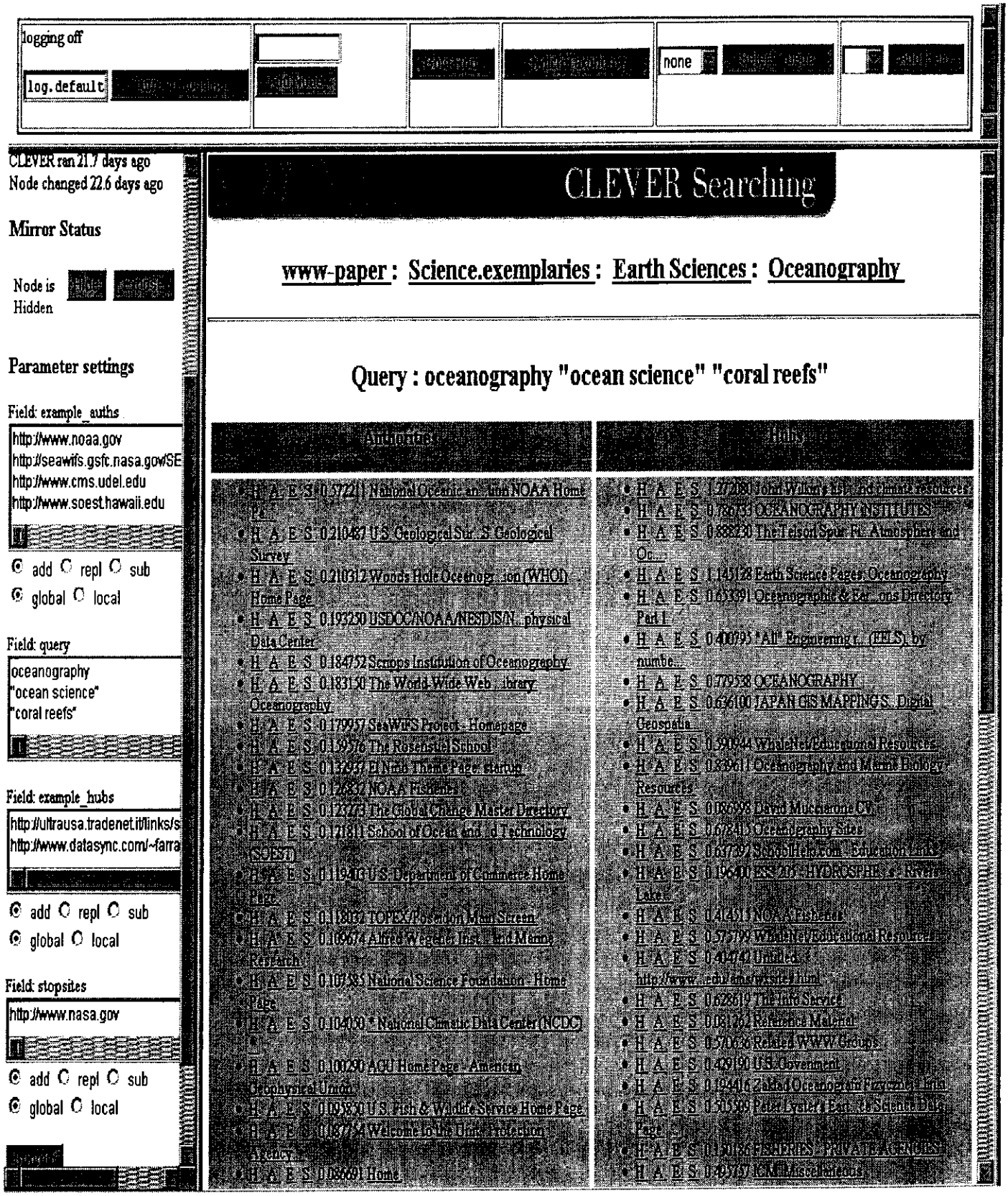

Figure 3.1: Screenshot of TaxMan.

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.

Building the taxonomies

Our experiment involved the construction of four taxonomies: three drawn from predefined subtrees of Yahoo! - Government, Recreation and 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" -navy -army "Scott Air Force Base" "Altus Air Force Base" "Barksdale Air Force Base" "Hanscom Air Force Base"). 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, The American solar energy society, The solar cooking archive, and Solarex and the example hub was Solar energy links. 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.

A number of points about our experimental setup are worth discussing 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 [CDG+98,CDK+98] 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, in the spirit of the work of Bharat and Henzinger [BH98] (who compare the relative results of eight variants they propose on the HITS algorithm of Kleinberg [K98].

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 pages at indices *h(N)* and *l(N)* in the list of hubs, and the two authority pages at indices *h(N)* and *l(N)* in the list of 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.

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 [BH98,CDG+98], 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 an html page containing links to four topics (the four topic nodes mentioned above), and was instructed as follows:

> Thank you for participating in our study. Several variants of the Clever system have been used to compile a topic tree, much as you would find in Yahoo and other portals. Please evaluate the following four nodes from this tree --- each node has up to 12 URL's, sampled from the results of different versions of Clever. For each tree node (topic) you rate, we would like you to judge each URL in the topic as "bad", "fair", "good" or "fantastic" (or "unreachable" if you cannot load the page) based on how useful the page would be in learning about the topic. A page may be useful either because it contains good content, or because it provides hyperlinks to good content. Your ratings will help us determine which variants of Clever yield good URL's.
>
> Please rate the following 4 topics. For each topic you rate, please be sure to judge all the URL's listed in that 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. The study was conducted in November 1998.

4. Results

In Section 4.1 we analyze our timing information to benchmark the ontologist effort required for each mode of construction, and give some preliminary statistics characterizing how our ontologists performed exemplification. Next, in Section 4.2 we analyze the raw performance of our three modes of taxonomy construction without considering the relative effort in each case. In Section 4.3 we combine user study results with logfile data to consider how each component of an exemplary query contributes to the final score of a node.

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. Finally, we offer some anecdotal evidence tying these scores and precisions values to earlier results; this anecdotal evidence suggests that the pages returned by our system represent high-quality results compared to some other sources.

4.1. Ontological effort

In this section we consider the amount of ontological effort required for the three modes of taxonomy creation. 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 wallclock 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 4.1.1. 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 25-40 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.

Time per node (seconds)

	Advanced	Exemplary
Science	108	119.76
Recreation	192.4	239.64
Personal	157.47	214
Government	270.39	222.43

Table 4.1.1: Average time to construct advanced and exemplary nodes.

Figures 4.1.2 and 4.1.3 also include summary information about the number of page visits our ontologists employed per node to gather information necessary to build the advanced and exemplified queries, the average time taken in each case, and in the case of exemplary queries, the number of hubs, authorities and stopsites. Note that the more specific personal taxonomy contains fewer exemplary pages than the more general taxonomies. Interestingly, we show below that although the overall results are worse for the personal taxonomy than for any of the other taxonomies, the benefits due to exemplification are significantly larger. This suggests that smaller communities containing only a few great hubs or authorities can be automatically populated more effectively if those few hubs or authorities are known.

Average statistics for advanced nodes

	Page visits	Average time
Science	0	108
Recreation	2.4	192.4
Personal	0.8	157.47
Government	2.9	270.39

Table 4.1.2: Average statistics for advanced nodes.

Average statistics for examplary nodes

	Example Auths	Example Hubs	Stopsites	Page visits	Av
Science	3.57	1.41	0.46	3.57	
Recreation	2.42	2.67	0.36	9.16	
Personal	0.89	1.53	0.05	4.32	
Government	1.96	1.36	0.25	4.36	

Table 4.1.3: Average statistics for exemplary nodes.

4.2. Results: Three modes of taxonomy construction

We begin by considering the average score of each mode of taxonomy creation. Unless otherwise specified, these averages are taken over both hubs and authorities; we examine the differences between hubs and authorities below.

Figure 4.2.1 shows a histogram showing the overall average of all evaluated links, by mode of creation. The number of judgments contributing to the average of each bar is between 400 and 500 in all cases. In Figure 4.2.2 we consider precision rather than average score, at various prefixes of the resource lists returned by Clever. As the figure shows, exemplification brings some improvement in average precision, particularly as we consider longer prefixes of resource lists from each node.

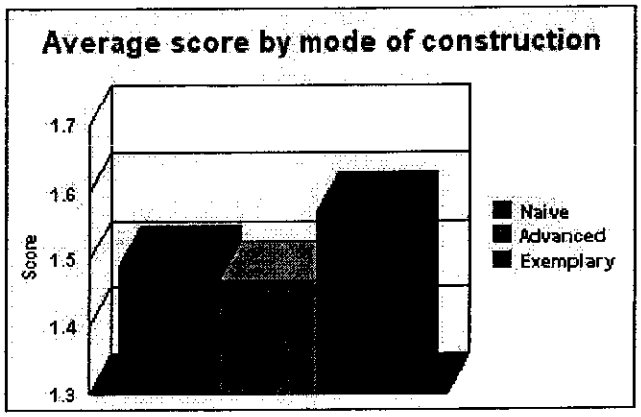

Figure 4.2.1: Average score of top 25 hubs and authorities, by mode of construction.

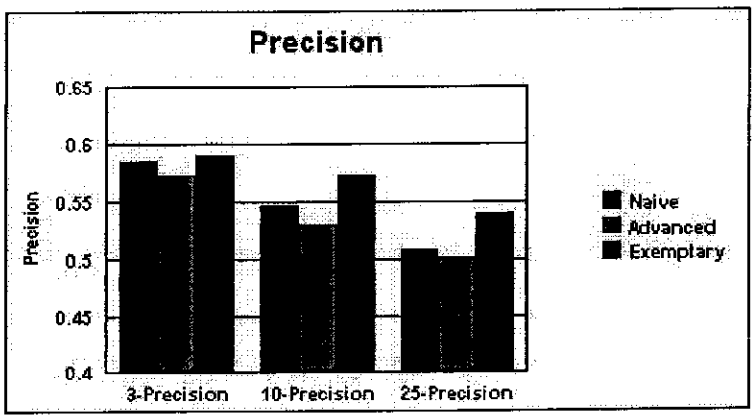

Figure 4.2.2: Precision at 3, 10, and 25, by mode of creation.

Differences among taxonomies

We now break the results of Figures 4.2.1 and 4.2.2 by taxonomy. Figures 4.2.3 and 4.2.4 show results for both average score and average precision at 25.

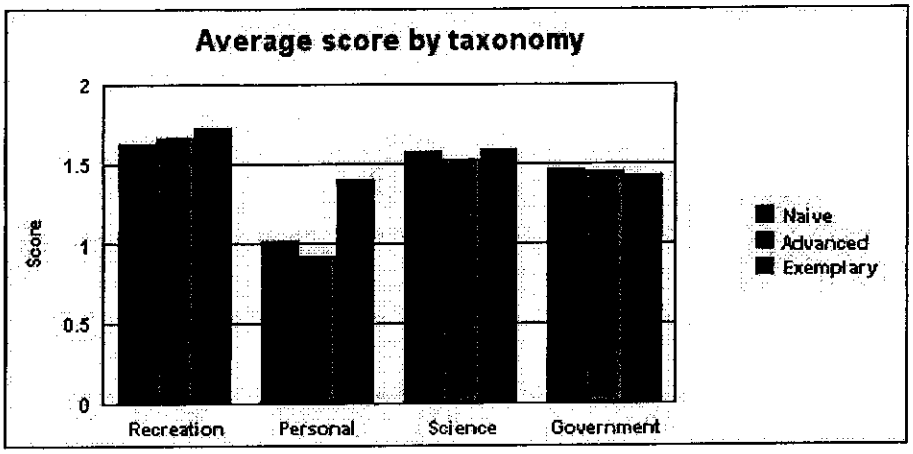

Figure 4.2.3: Average score of top 25 hubs and authorities, by taxonomy.

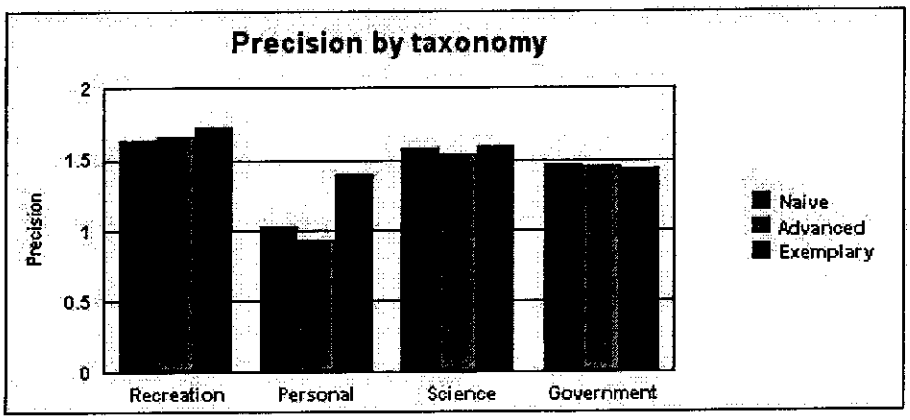

Figure 4.2.4: Average precision of top 25 hubs and authorities, by taxonomy.

As expected, performance differs according to the taxonomy. As noted above, the personal taxonomy was much more difficult than the general-interest taxonomies; an examination of the nodes shows that topics in the personal taxonomy tend to be narrower in focus. For instance, the following are some of the nodes under Personal : Work

Computers : Computing Companies
Conferences : FOCS/STOC
Conferences : SIGMOD
Conferences : WWW
Information Retrieval : Collaborative Filtering
Information Retrieval : Latent Semantic Indexing
Information Retrieval : Phrase Extraction
Programming Tools : C
Programming Tools : Java
Programming Tools : Perl

Tex : Latex
Theory : Crypto : Kerberos
Theory : Crypto : 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.

We continue examining the differences between taxonomies by considering the fall-off of result quality as we include pages from farther down the ranked list. Figure 4.2.5 shows the results of Figure 4.2.1 broken by index of page; that is, the figure 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." Each point in these curves represents the average of at least 30 samples. The figure confirms our earlier remarks regarding the difficulty of the personal taxonomy. Average score falls off towards the tail of the list faster than it does for the general-purpose taxonomies.

The government taxonomy shows high scores for the first two indices, and then performance lags the other general taxonomies. The occurs because many nodes in the government hierarchy have a single "appropriate" authoritative response. The department of education node, for instance, has an obvious top authority in the department's homepage, but it is difficult to find a large number of equally authoritative sites.

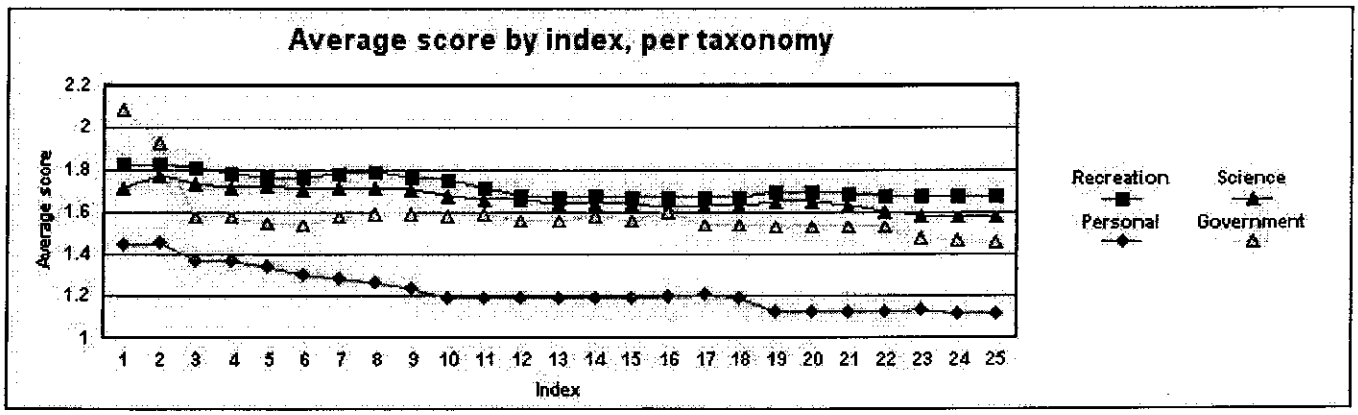

Figure 4.2.5: Average score by index for each taxonomy.

Performance by index

We now consider the same graph of average score versus index, broken now by mode of creation rather than taxonomy. As Figure 4.2.6 shows, naive and advanced text queries perform similarly, and exemplification allows a significant improvement in performance both early and late in the resource lists. Again, each point in these curves represents the average of at least 30 samples.

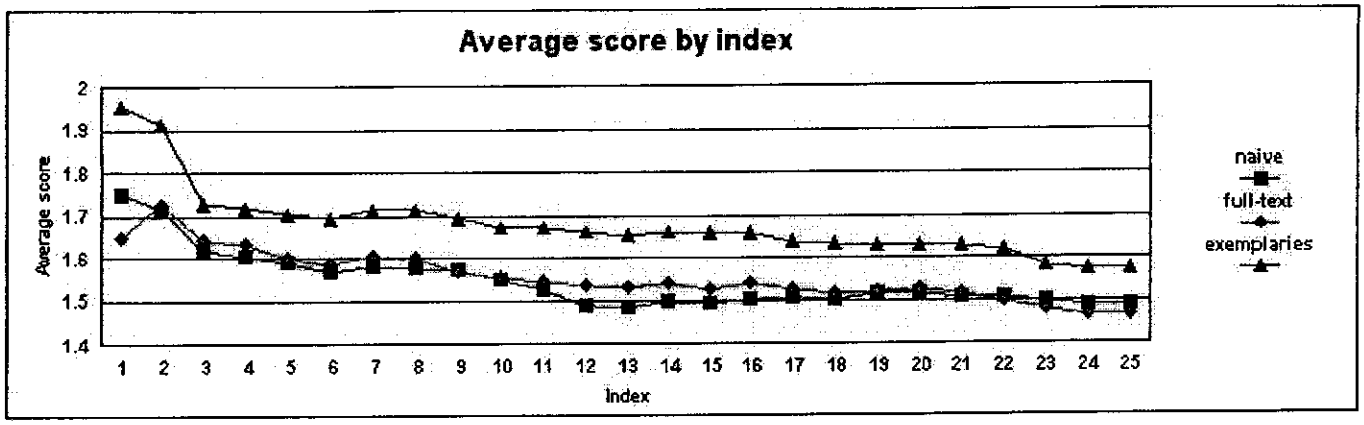

Figure 4.2.6: Average score by index for naive, full-text, and exemplified queries.

Improving lower-quality nodes

We now consider the improvement (from advanced queries and exemplification) at a node as a function of the performance of the naive query at that node. Figure 4.2.7 shows the improvement over naive queries from a full-text query, and likewise Figure 4.2.8 shows the improvement over full-text queries from exemplification. As the figures show, these techniques are effective at improving the quality of a poor node, but are ineffective or deleterious for nodes that already contain good results. Clearly, the improvement in quality resulting from exemplification is the least (and in fact sometimes negative) when the naive query does especially well. This is not a contradiction. As mentioned above, our goal is to study incremental techniques for taxonomy creation: specifying a naive query, then improving the results if necessary using more powerful and time-consuming tools. Therefore, in creating the taxonomies, we forced ourselves to enter full-text and example queries even if we thought the naive query had done such an excellent job that no further exemplification was needed. In practice, we expect that an ontologist who has reaped a good harvest of URL's from a naive query would not bother to go to the length of advanced text queries, or exemplification.

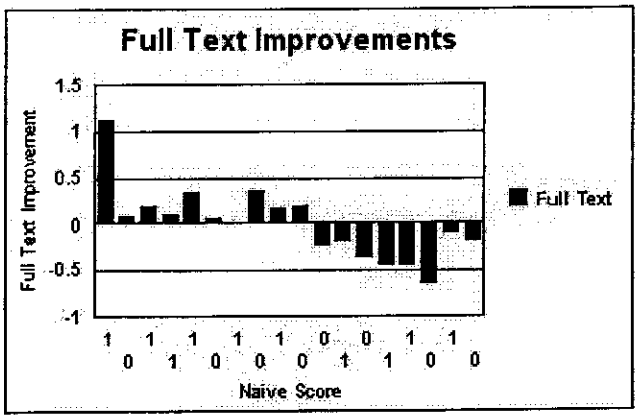

Figure 4.2.7: Improvements from Naive to Full Text modes of construction, by naive score.

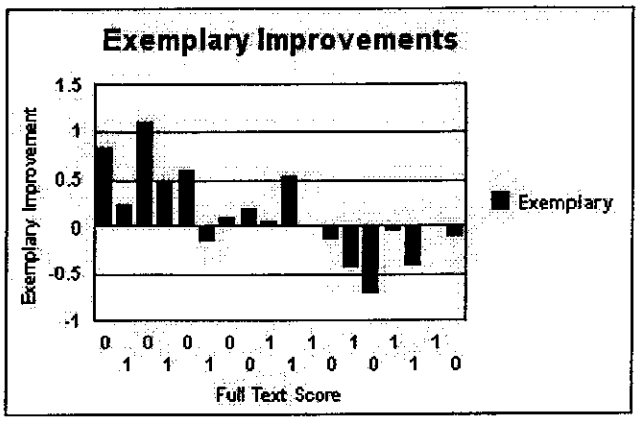

Figure 4.2.8: Improvements from full text to exemplary modes of construction, by full text score.

Hubs versus authorities

We now present the by-index results of Figure 4.2.6 broken into hubs and authorities and averaged over all modes of creation. Most importantly, the authorities are significantly better-rated than the hubs. In all prior studies of the effectiveness of HITS-based systems, hubs have scored better than authorities. The algorithmic developments described above in the use of example authorities have not been studied before, and are particularly focused on retaining purity in the authority list; this may be responsible for the shift.

The authorities also show a peak at index two. Due to the nature of our taxonomies, we often consider nodes on a general topic with a set of perhaps five to ten subtopics. If the general topic is Astronomy, the subtopics might be Clubs, Observatories, and so forth. There is often a single obvious authority appropriate for the parent node, such as NASA in the case of astronomy. The children often rank this obvious parent as the first authority, and then go on to list more specific resources about the subtopic. This results in the spike of the figure.

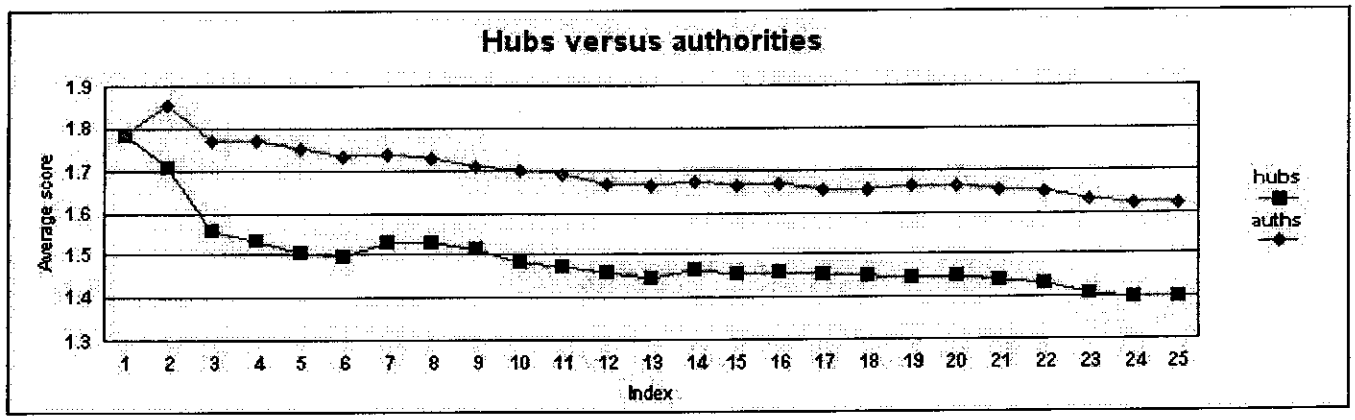

Figure 4.2.9: Hubs versus authorities.

Final remarks on page quality

To put these results in context, we now give some anecdotal support for the quality of the pages returned by the system in all cases. Using the binarization of our ratings as defined above, we compute the precision of the top 10-ranked pages to be .57 (see Figure 4.2.2). In earlier work [CDG+98] we found using the same evaluation metric that the average precision for Yahoo was .38 and for earlier versions of Clever was .48. Note that the absolute magnitude of these precision values is low, reflecting our stringent definition of relevant. If we took pages rated "fair" to be relevant, all three precision values would increase significantly. Further, the results once again point to the fact that users are judging quality as opposed to relevance; Yahoo pages are almost certainly relevant to the topic, but might not be the best available. Finally, the studies used different users, drawn from a similar population.

4.3 Effort versus quality

We begin by considering the number of links enrolled as example hubs and authorities. Figure 4.3.1 considers average page score as a function of number of example hubs and authorities. Figure 4.3.2 likewise shows page score as a function of number of stopsites, time spent at a node, and number of "page visits", or excursions taken from TaxMan in order to explore other material. These figures support a number of anecdotal lessons that we have discovered as ontologists in the process of exemplifying pages:

- if it is possible to find at least one good hub, performance is likely to improve dramatically
- too many example hubs or authorities begin to degrade performance
- if spurious pages arrive and require stopsiting, this suggests that Clever may be unable to find a clear community around the topic.
- nodes that are more difficult to specify correctly (as evidenced by time or number of page visits) tend to induce more highly varying performance of the algorithm.

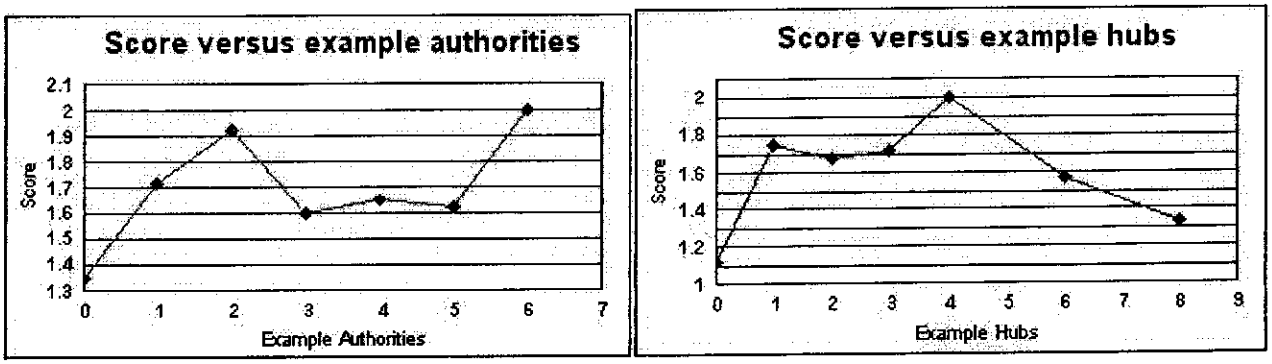

Figure 4.3.1: Score as a function of number of examples.

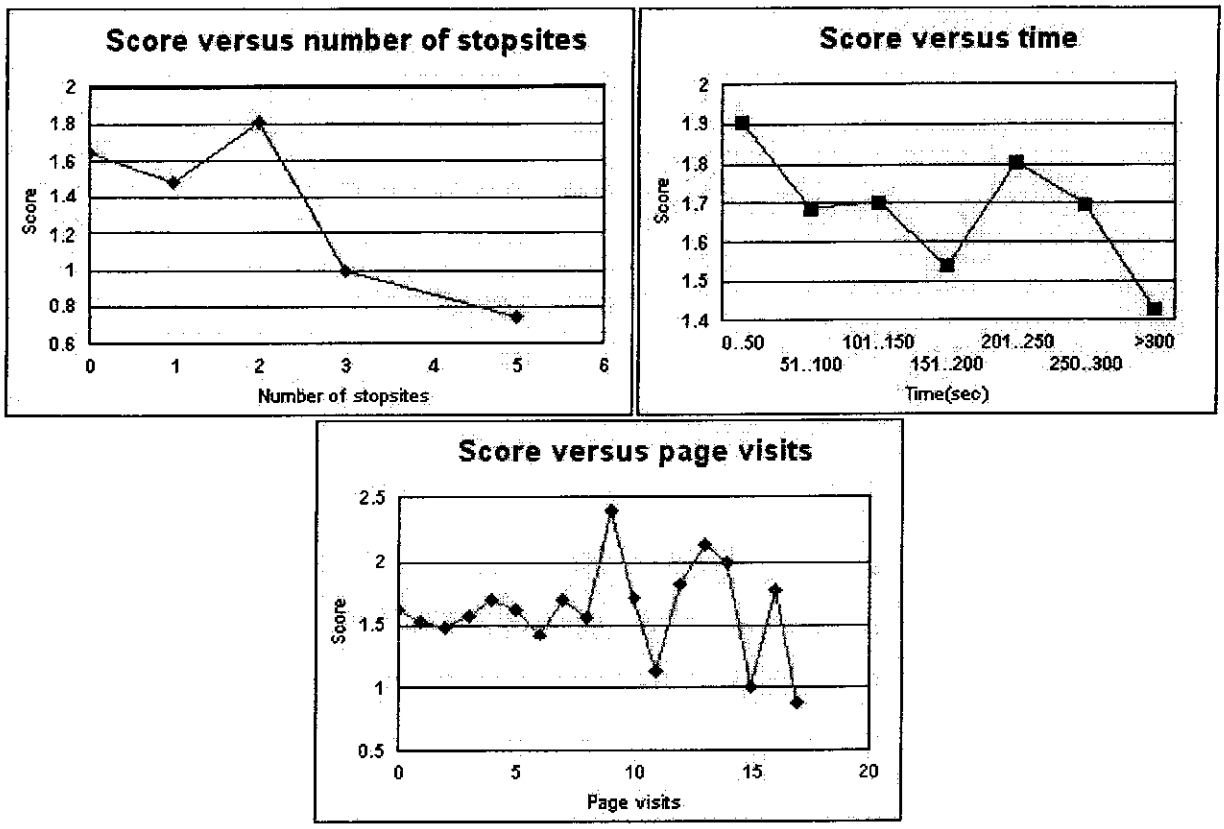

Figure 4.3.2: Score as a function of number of stopsites, time, and page visits.

5. Conclusions

We have presented extensions to the Clever system for semi-automatic taxonomy construction, and described and benchmarked a sample 450-node taxonomy created using Clever. Constructing such a taxonomy involves finding good URL's, culling the best among them, and annotating them; we mainly focus on the first two of these activities. In the process of building the taxonomy, we describe three paradigms for topic searching. We show via a user study that an ontologist, armed with these paradigms, can create a high-quality taxonomy with a fairly quick turnaround time.

References

BH98

K. Bharat and M.R. Henzinger. *Proceedings of ACM SIGIR, 1998.*

CDG+98

S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. Proceedings of the *7th World-Wide Web conference*, 1998. Copyright owned by Elsevier Sciences, Amsterdam.

CDK+98

S. Chakrabarti, B. Dom, Ravi Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins.

Experiments in Topic Distillation. *SIGIR workshop on hypertext information retrieval*, 1998.

K98 J.M. Kleinberg. Authoritative sources in a hyperlinked environment. Proc. *9th ACM-SIAM Symposium on Discrete Algorithms*, 1998. Also appears as IBM Research Report RJ 10076, May 1997.

R79 C.J. van Rijsbergen. Information Retrieval. Butterworths, 1979. See also here.