Cooperative Indexing, Classification, and Evaluation in BoW

Dror G. Feitelson

School of Computer Science and Engineering,
The Hebrew University, 91904 Jerusalem, Israel,
feit@cs.huji.ac.il,
WWW home page: http://www.cs.huji.ac.il/~feit

Abstract. BoW is an on-line bibliographic Dynamic Ranked Information-space (DyRI). It provides the infrastructure for users to add bibliographical information, classify it, index it, and evaluate it. Thus users cooperate by contributing and sharing their experience in order to advance the most problematic aspects of information retrieval: finding the *most relevant* and *high quality* information for their needs.

1 Introduction

The basic problem in information retrieval today is filtering the massive amounts of information that are available in order to find high-quality relevant information. The quest for high quality means that the available information must be evaluated and ranked in some way. The quest for relevance means that the information must also be classified and indexed according to pertinent concepts.

Current information retrieval systems often leave much of this filtering to the users. They focus on an effort to be comprehensive, producing a superset of the desired information. The user then shifts through this information, discarding most of it, and selecting those items that seem to best answer the needs. But the effort expanded in this selection process — in which *a human user with understanding of the domain checks the system's classification and performs an evaluation* — is lost. The system does not keep track of which data items were selected in the end, and does not have the means to match them with a refined version of the user's original query.

The BoW project is an attempt to investigate the possibility of tapping the work done by users to improve the system. The scope chosen is a bibliographic repository for a limited domain (BoW stands for "Bibliography on the Web", and our prototype contains approximately 3000 entries from the domain of parallel systems). Within this scope, users are provided with facilities to contribute to the classification and indexing of entries, and the same facilities are used for the incremental construction of queries. In addition, the system keeps track of users' searches and their results, and uses this information to reorganize the way data is presented to subsequent users. Thus valuable user experience contributes to improving the system's service, rather than being lost.

2 Dynamic Ranked Information Spaces

2.1 The Vision

Consider a situation where you are a university professor specializing in parallel systems, and one of your students comes to you with an idea for a new network topology. You recall that you have seen something like this in the past, but you do not remember the name given to this topology or who did the work. An altavista search using the term "network topology" produces 12,338 hits, and those you check either describe specific installations or are dangling links pointing to nothing. Your only recourse is to try and call up some colleagues who might have a better memory.

Now consider what might have happened if the parallel processing community maintained a dynamic ranked information space with technical publications in this field. You would enter at the root node, and traverse the path "architecture" $\rightarrow$ "interconnection networks" $\rightarrow$ "topologies" to arrive at a page listing hundreds of proposed topologies, grouped according to their attributes. For each one you will be able to get a concise description, the text of research articles describing the topology and its uses, commentary on these articles, links to descriptions of systems that actually use this topology, and an indication of how many other researchers are also interested in it. If you find that any of this information is stale or misleading, you will be able to either leave a comment about it, or alert an editor that a link should be removed. Thus your experience will immediately contribute to the maintenance of the site, as the experiences of others have contributed before you.

This example is not unique to parallel systems or even to searching in the scientific literature. For example, a similar situation can occur with an architect looking for data on designing public libraries in a dynamic ranked information space dedicated to that topic. The basis is the existence of a tightly knit community of users that contribute to the maintenance and updating of the repository by submitting information, commentary, and suggestions for structural changes. As a result, the repository changes dynamically with time (rather than just accumulating more and more items), and contains feedback and evaluations in addition to the original raw data items. In addition, using the repository shortens the publication time of new information to zero, and makes it available in multiple cross sections. It is a large scale extension of the concept of peer review, coupled with an indexing service.

The project follows the "field of dreams" approach, which is actually the basis for the growth of the Internet [5]: we just *provide the technology* for creating the information space, and *leave it to the users to supply the content*[1]. The resulting system is called a "**Dy**namic **R**anked **I**nformation-space", or DyRI for short.

[1] However, initially it is necessary to prime the information spaces with enough content to make them sufficiently attractive so that potential users overcome the "new technology" barrier.

2.2 The Design

While the concepts explored here apply to any information repository, we use a bibliographic repository for concreteness. This also simplifies the prototype by limiting it to rather structured data.

User Types Users of DyRIs are classified into three types: users, contributors, and editors.

Users are those who use the information space to search for information. The main search method is by traversing a concept index that classifies the available information according to content. This allows for refining the search as one proceeds, rather than requiring one to have a clear notion of the required information at the outset.

While general users do not add information to the information space, they do register feedback relating to existing data. One form of feedback is simply by traversing the concept index: the system keeps counts of visits to each page, and uses this information to identify the more popular ones to future users. In addition, users may register positive or negative feedback to each page, to note their level of satisfaction. Again, the system displays this information as part of the indication of a page's popularity.

Contributors are users who not only search for data, but also contribute data. In principle any user may become a contributor; the only requirement is to identify oneself to the system. Such identification is required both in order to attribute contributions such as annotations to their authors, and in order to identify the arguing parties in case of disputes. In extreme cases of misuse, it may be necessary to limit certain users.

Contribution can take any of three forms:

- Adding a new entry to the repository.
- Adding an annotation to an existing entry, providing additional insight into its importance or content.
- Adding a link between related pages or entries.

Links create the fabric of the concept index, allowing it to be traversed incrementally, at the same time narrowing the scope of the search. Whenever a new entry is added to the repository, it should be linked to appropriate pages in the concept index. Contributors who discover additional meaningful links later may also add them.

In addition, contributors can make minor modifications to existing data, e.g. in order to correct errors. Contributors can also *suggest* major modifications, such as deleting entries or links. However, acting upon such suggestions is left to *editors*, after proper solicitation of a rebuttal from the original contributor. Thus the editor's main task is to resolve conflicts and maintain the quality of the repository, based on input from the contributors.

The Concept Index Most search engines are unsatisfactory because users are required to have a good notion of what they are looking for before they start. The most common approach is to describe the query using keywords and logical operations; for example, the query (scheduling & (parallel | distributed)) is read as "find documents including the word 'scheduling' and either of 'parallel' or 'distributed' ". The intent is probably to find references regarding scheduling in parallel or distributed systems. However, the issue of whether we mean job scheduling by the operating system, task scheduling by the runtime system, or task scheduling by the compiler is left open. A good search engine will find all three types (and maybe more) and leave it to the user to shift through them. Adding keywords can reduce this burden, but runs the risk of false negatives, where items of interest are rejected because they do not contain all the specified

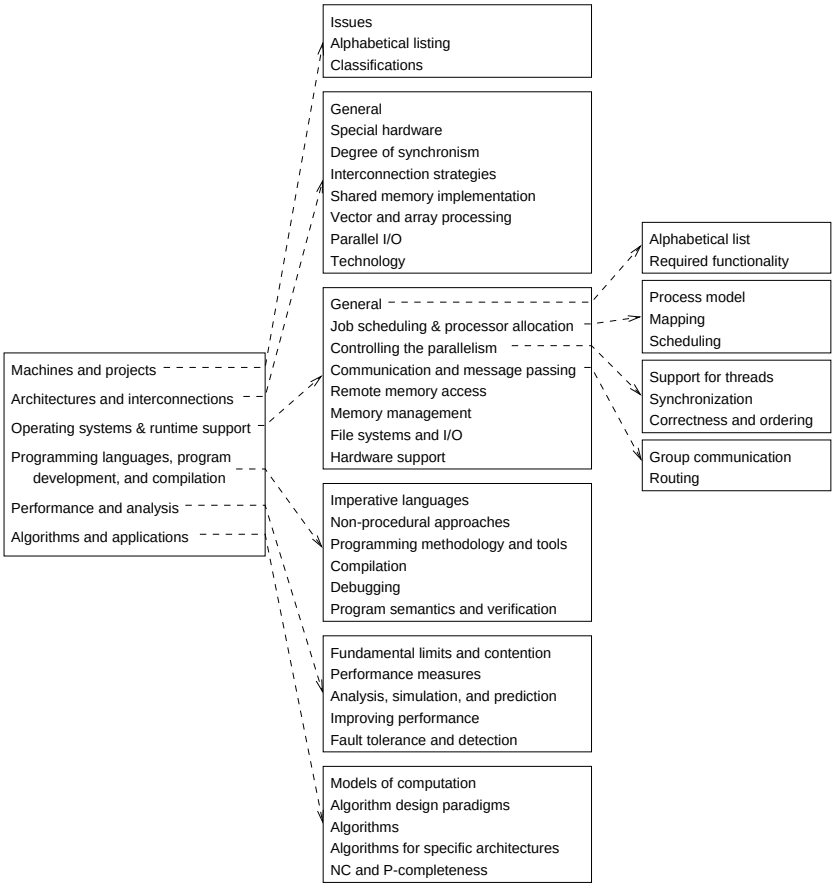

Fig. 1. *Example of the top levels of a concept index for an information space on parallel systems.*

keywords (and without any typos 8-).

In contrast, the concept index allows users to formulate their search incrementally on-line, and does not depend on matching keywords. Essentially it can be viewed as a menu-driven search. The top level of the index (the root) contains links to several broad topics. Following such a link leads to a page representing the chosen topic (a *concept page*), and including a list of subtopics and/or bibliographic entries. The index is navigated using a hypertext interface such as a Web browser, by going from one concept page to another. The leaves contain only bibliographic entries that pertain to a narrow and focused topic.

An example of the top levels of a concept index for the domain of parallel systems is shown in Fig. 1. Using such a structure, a user looking for information on the scheduling of parallel jobs will follow the "operating systems and runtime support" link from the top level, and then the "Job scheduling & processor allocation" link. A user looking for information about on-line task scheduling would diverge at the second level, and choose the "controlling the parallelism" link. A user looking for information about task scheduling by the compiler would start with the "programming languages, program development, and compilation" link at the top level, and then choose "compilation".

As noted above, the structure of the concept index is of utmost importance. For any specialized domain, it seems advisable to create a special index based on a thorough understanding of the domain. This should be done with an eye for what users might look for. The topics need not be (and probably should not be) completely disjoint: the index structure can easily be a DAG rather than a tree. Thus any subtopic that is relevant to two or more larger topics (e.g. if it represents their intersection) is simply linked to all of them, and can be found by several distinct routes in the index. For example, it would be convenient if information on "virtual memory" was accessible both via "architectures" $\rightarrow$ "shared memory implementation" and "operating systems" $\rightarrow$ "memory management".

An important question is the "right" size for pages, and the resulting depth of the index [2]. The tradeoff is between scrolling and loading. Using small pages that do not require scrolling leads to a deeper index, and therefore requires more pages to be loaded from the server. If we want to reduce the average number of pages loaded in order to reduce the accumulated waiting time, we need to use larger pages. A possible way out is to use a relatively low branching factor, but show *two levels* of the index in each page. The pages are then bigger, but their internal structure makes them easier to use.

User Feedback An important part of the interface is the support for registration and display of user feedback. The feedback feature is embedded naturally into the concept index, so as to be usable and useful without any training. Registering feedback about links in the concept index is done as a byproduct of traversing these links. One simple form of feedback is popularity: the system keeps count of the number of times that each link is traversed, and displays this *at the head of the link* (rather than displaying a count of visits in the page itself). Note that if a page has more than one link pointing at it, the counts for

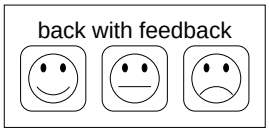

Fig. 2. *Composite "back" button used to obtain feedback.*

these links will be different, as they well should be, because they represent the perceived relevance of the page *in different contexts*.

The problem with mere counts is that they represent the *initial perception* of users, but not their *final satisfaction with their choice*. To capture user satisfaction, pages have composite "back" buttons embedded in them (Fig. 2). By selecting the happy or sad face, the user can distinguish between a "happy back after finding what I wanted" and a "frustrated back after failing". Clicking elsewhere just performs the back function, without registering any feedback.

The problem with the composite back buttons is that we do not want to burden the user with them. Therefore *inferential* feedback is used as well. One form of inferential feedback is that positive feedback is applied to the whole path from the root to the page on which the happy face is pressed, thus saving the need to go all the way back to register satisfaction. Negative feedback, on the other hand, is applied only to the last link, allowing for backtracking and trying of other links. Another form of inferential feedback is that using the "export" facility is deemed to represent positive feedback, based on the assumption that the user is exporting data because he likes what he found.

Once the system has the feedback information, it should display it in a useful manner. The suggested approach is to decorate each link with a small icon that presents the information graphically. Specifically, a set of marks can be used, with green check-marks denoting positive feedback and red X's denoting negative feedback. The number of marks indicates the degree of positiveness or negativeness, while their size reflects the total number of visits. Such a display allows users to focus immediately on "big check-mark" links, which are those that many other users have found useful.

The specific formula used combines the ratio of good to bad feedback with a logarithmic scale, so as to allow for a large dynamic range. The formula for the number of check-marks n is

$$n = \lg\left(\frac{g-b}{b+1} + 0.7\right) + 1$$

where g and b are the numbers of good and bad feedbacks, respectively. For X's, exchange g and b. This leads to numbers as indicated in Fig. 3.

An additional use of feedback is the internal organization of the concept pages. It is envisioned that at lower levels of the index concept pages will be divided into topics, each listing a set of relevant bibliographic entries. The order of entries in such a set should be

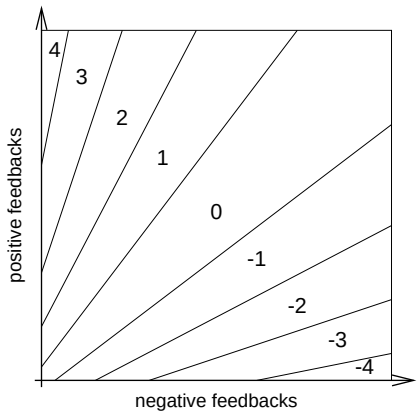

Fig. 3. *dependence of feedback visualization on actual number of positive and negative feedbacks.*

1. New documents that were only recently added to the repository. Such documents are kept on top for a few months or until they get some feedback from users.
2. Documents that have received positive feedback.
3. Documents about which users are ambivalent, or no feedback is available, possibly for lack of popularity.
4. Documents that have received negative feedback. These documents are candidates for removal.

3 Comparison with Other Approaches

The issue of finding relevant information according to one's needs is obviously of paramount importance. It has therefore motivated considerable research and development activity, and the creation of a large industry that indexes and provides access to on-line information. There are three main approaches: using links, using keywords, and using experts.

3.1 Finding Information Using Links

The Science Citation Index is based on the notion that related papers in the scientific literature are "linked" by their bibliographical citations: either they cite each other, or they share many citations. Thus if you have a starting point for your search, namely some scientific paper on this topic, you can use both *its* citations and citations *to it* to find additional related papers. Citations in a paper are easy to find: they appear in the paper itself. The Science Citation Index provides the other direction: for any given paper, it lists other papers that cite it.

This idea has been extended in two ways using the WWW. One is to make references into hyperlinks, and collect on-line documents that reference each other. An example is the NEC Research Index (http://csindex.com), which is based on an automatic tool that crawls the Web looking for scientific papers, analyzes them, and creates a citation index from them [4]. A related example is the hypertext bibliography project (http://theory.lcs.mit.edu/~dmjones/hbp). This project contains information about all papers published in a host of journals and conferences, mainly related to Theoretical Computer Science, and maintains two-way links among these papers according to their citations.

The other extension is to apply these ideas directly to the structure of the WWW itself. The thrust of the work in this direction is based on Kleinberg's classification of Web sites into hubs and authorities, based on the the links emanating from them and pointing to them [3].

3.2 Finding Information Using Keywords

Indexing services are another well-known search facility. They don't require you to have a starting point — only a good notion of what you are looking for. They survey the scientific literature as it is published, and collect papers by topic. Thus they provide a much needed mapping from topics to journal pages, much as an index at the end of a book provides a mapping from topics to the pages of the book.

The enormous quantity of information that is available, and its exponential growth rate, has lead to much interest in automatic indexing [7]. Direct improvements to simple indexing include the ability to derive related words, and knowledge about synonyms based on a thesaurus [1, 8]. Using these facilities, it is possible to find useful information even in cases where a direct match to the user's query does not exist.

A more sophisticated approach is based on learning from examples using boosting algorithms [6]. These algorithms combine multiple inaccurate heuristic classifications (e.g. based on keywords) into a more accurate final classification (e.g. a prediction of the topic being addressed in the document). The methodology involves iterative learning using pre-classified examples. Each example attaches a set of labels to a document. The system learns to attach such labels automatically, by iteratively refining its notion of how combinations of inaccurate classifications lead to final classifications. This in turn is based on giving higher weights to those examples that are hardest to classify.

3.3 Indexing by Experts

While machine learning can help achieve good classifications based on keywords, some believe that ultimately there is no alternative to human understanding. Indeed, most Internet portals now include large indexes (usually called Internet Directories) maintained by their staff in addition to the traditional keyword search facility (Table 1). These indexes classify the whole Internet according to a hierarchical structure, and provide lists of generally useful web pages for each

Table 1. *The sizes of major Internet directories at end of 1999, according to Search Engine Watch (*http://www.searchenginewatch.com*).*

directory	*editors*	*categories*	*links*
Yahoo!	100+	?	1200000+
LookSmart	200	60000	1000000
Open Directory	15400	153000	950000
Snap	30–50	64000	600000

topic. The Yahoo! index is especially interesting, as it is a DAG rather than a tree, with explicit indication that some pages are shared by several branches (http://www.yahoo.com). The Open Directory (whose slogan is "HUMANS do it better") also has a very useful structure: The home page displays two levels of the directory, with links to pages on the main topics, and below each one, links to more focused subtopics. About.com (http://www.about.com) is a network of sites maintained by experts in various fields, which makes a point of parading the human experts rather than the technology.

3.4 Problems and Comparison

There are various problems with the abovementioned approaches. One is that many of them *lack interpretation*. This means that papers are associated according to superficial attributes (citations or keywords), not according to an understanding of what the papers are actually about and how they relate to each other. There is no real editorial work on classification and organization. Moreover, users cannot augment these mechanisms with private annotations that do contain interpretation.

Another is that they are often *source oriented*. This relates to the choice of papers that are covered: certain journals are selected, and all the papers that appear in them are included, starting from a certain year. Granted, an effort is made to select as many journals as possible, and to focus on the best journals, but economic and business considerations may sometimes prevail over technical ones. Moreover, even good journals sometimes contain not-so-good papers, and some good papers are published in obscure journals. Results that are only published in conferences or technical reports are excluded outright. So are old papers that were published before the indexing commenced.

A third problem is that of *coverage and quality*. This problem is especially common in keyword-based search, where hits that do indeed contain the requested keywords have widely different levels of importance and usefulness. In the extreme case we have false positives, which contain the desired keywords but are totally irrelevant. An example is a search for "gang scheduling"[2] which retrieved a web page that included the sentence "The RV6 forum got off to a

[2] A scheduling technique used on parallel computers whereby a job's processes are scheduled simultaneously on distinct processors.

rocky start due to a scheduling misunderstanding with the Van's gang". A related problem is false negatives, that is relevant and useful documents that use synonyms or related terms are therefore not found.

The problem with human experts is that they are expensive, so there is a necessary tradeoff between the number of experts and the size of the fields that they have to cover. As a result, the human classifiers cannot in general have cutting edge knowledge about all their fields.

The BoW project is based on the idea that indexing by paid experts is futile. Instead, indexing and ranking must be done by the users of the information, thus tapping their enormous combined pool of knowledge and experience. Of special importance is the support of ranking and evaluation of documents, which does not exist in other projects. It is this ranking which counteracts the exponential growth of information, and ensures that high-quality information becomes more visible. The thrust of our work is to create the infrastructure and technology to enable such a mode of cooperation.

4 The BoW Prototype

The BoW project has been ongoing for a couple of years, and two generations have been completed. An example screen dump of a page from the concept index of a parallel systems information space is shown in Fig. 4. The prototype supports insertion of new bibliographic entries, addition of annotations, creation of links from concept pages to entries, among entries, and to external web pages (thus supporting the publication of full text rather than only references), user feedback and display, and exporting of bibliographical entries. It has a concept index of 142 pages, in which 8201 links to entries are grouped according to 3167 topics. In all, there are 3046 entries, for an average of 2.7 links per entry. This is all based on an automatic conversion of a bibliographic database kept in LaTeX/BibTeX format since 1988. It can be accessed on-line at URL http://www.bow.cs.huji.ac.il.

The implementation is based on using perl mode in an Apache Web server. The concept index is mirrored in a directory hierarchy, and concept pages are generated on-line as required by reading the appropriate directory. Thus all updates and changes appear automatically once the underlying directory structure is modified.

Several problems arise from the fact that the http protocol is stateless, and provides only limited support for continuous sessions (using cookies). Currently this causes problems with collecting a list of entries that should be exported upon demand; when user registration is implemented, we will also need to keep track of the user. The initial solution was to send the whole export list back and forth in each transaction. The second version improved on this by keeping the list in a memory segment shared by all the httpd processes in the server, and only sending a session ID in each transaction.

Features that are now being implemented as part of the third generation include

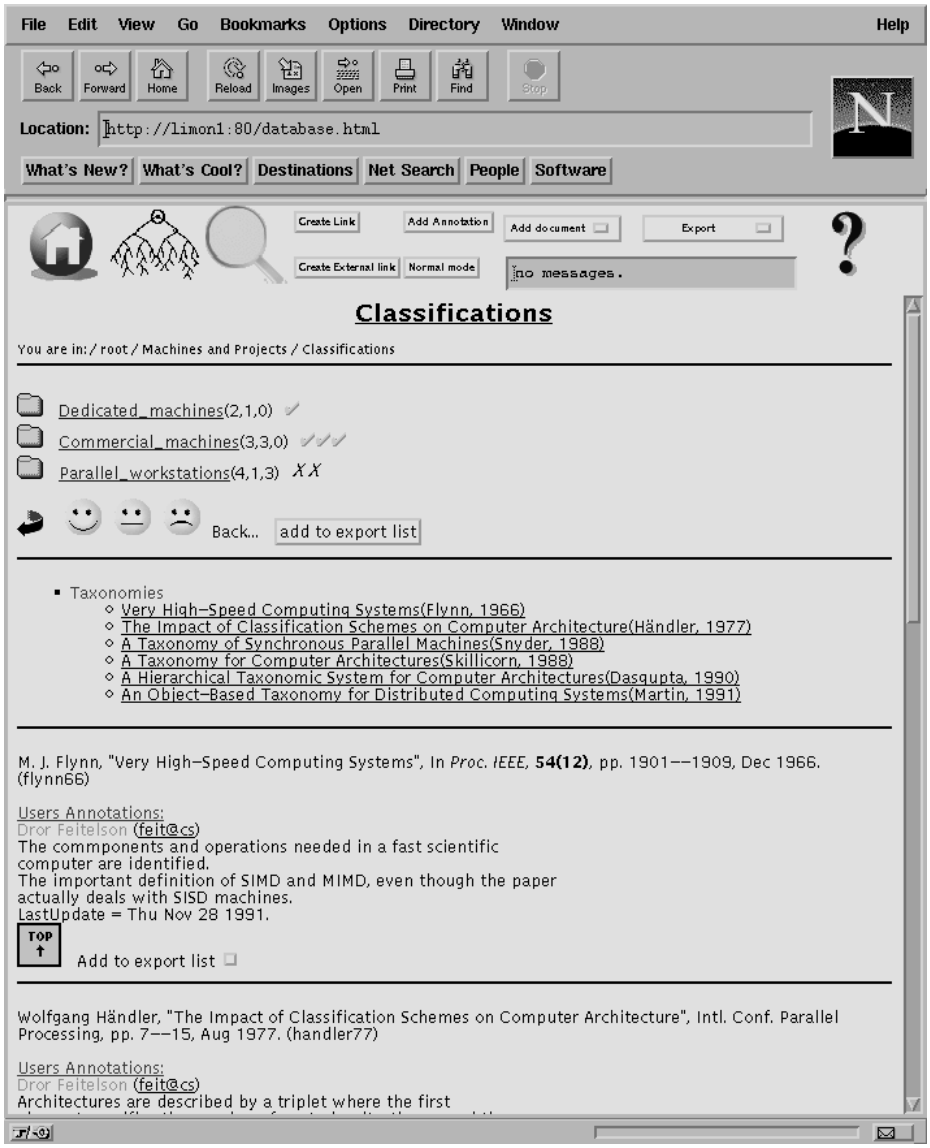

Fig. 4. *Example of concept page from the prototype information space on parallel systems. It includes links to sub-topics with an indication of user feedback, a listing of entries that belong in this page, and then the entries themselves, including bibliographic information and annotations.*

- Mechanization of the citation format to enable better identification of duplicate entries. In particular, journal and proceedings titles should come from a menu.
- Control over the structure of the concept index with an XML-based format.
- Provision of an automated listing of suggested concept pages, where a newly inserted entry may be indexed. This is generated based on similarity between the new entry and entries that are already indexed and linked to these pages.

5 Conclusions

The main idea behind DyRIs is that *users* can and should cooperate to improve the quality and usefulness of an information space. We designed and implemented one way of doing so, which is based on minimal active participation by users: they are invited to (but not forced to) add annotations and links to concept pages, and can provide feedback by using back-with-feedback buttons. This was a basic design decision, based on the fear that more extensive features such as feedback forms will go unused. We hope that the annoyance for users with our minimal design will be small enough that they will actually use these features. We intend to make the prototype information space on parallel systems publicly available once version 3 is ready, in order to test it in a real world setting.

Acknowledgements

This research was supported in part by the Ministry of Science. The first versions of the prototype were implemented by David Er-El and Roy Peleg.

References

1. H. Chen, J. Martinez, A. Kirchhoff, T. D. Ng, and B. R. Schatz, "*Alleviating search uncertainty through concept association: automatic indexing, co-occurrence analysis, and parallel computing*". *J. Am. Soc. Inf. Sci.* **49(3)**, pp. 206–216, 1998.
2. S. H. Kim and C. M. Eastman, "*An experiment on node size in a hypermedia system*". *J. Am. Soc. Inf. Sci.* **50(6)**, pp. 530–536, 1999.
3. J. M. Kleinberg, "*Authoritative sources in a hyperlinked environment*". In 9th *ACM-SIAM Symp. Discrete Alg.*, pp. 668–677, Jan 1998.
4. S. Lawrence, C. L. Giles, and K. Bollacker, "*Digital libraries and autonomous citation indexing*". *Computer* **32(6)**, pp. 67–71, Jun 1999.
5. R. W. Lucky, "*New communications services — what does society want?*". *Proc. IEEE* **85(10)**, pp. 1536–1543, Oct 1997.
6. R. E. Schapire and Y. E. Singer, "*BoosTexter: a boosting-based system for text categorization*". *Machine Learning* **39(2/3)**, pp. 135–168, May 2000.
7. B. R. Schatz, "*Information retrieval in digirtal libraries: bringing search to the net*". *Science* **275(5298)**, pp. 327–334, Jan 17 1997.
8. L. W. Wright, H. K. Grosetta Nardini, A. R. Aronson, and T. C. Rindflesch, "*Hierarchical concept indexing of full-text documents in the unified medical language system information sources map*". *J. Am. Soc. Inf. Sci.* **50(6)**, pp. 514–523, 1999.