A Unified Strategy for Search and Result Representation for an Online Bibliographical Catalog

Maayan Zhitomirsky-Geffet,
Department of Information Science, Bar-Ilan University, Ramat-Gan, Israel.
Email: maayan.geffet@gmail.com

Dror G. Feitelson,
School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel.
Email: feit@cs.huji.ac.il

Eitan Frachtenberg,
School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel.

Yair Wiseman
Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel.
Email: wiseman@cs.biu.ac.il

Category: ***Research paper***

Structured Abstract

Purpose:

One of the biggest concerns of modern information retrieval systems is reducing the user effort required for manual traversal and filtering of long matching document lists. Thus, our first goal in this paper is to propose an improved scheme for representation of search results. Further, we explore the impact of various user information needs on the searching process aiming to find a unified searching approach well suited for different query types and retrieval tasks.

Design:

The BoW on-line bibliographic catalog is based on a hierarchical concept index to which entries are linked. The key idea is that searching in the hierarchical catalog should take advantage of the catalog structure and return matching topics from the hierarchy, rather than just a long list of entries. Likewise, when new entries are inserted, a search for relevant topics to which they should be linked is required. Therefore, a similar hierarchical scheme for query-topic matching can be applied for both tasks.

Findings:

However, our experiments show that different query types used for the above tasks are best treated by different topic ranking functions. To further examine this phenomenon we conducted a user study, where various statistical weighting factors were incorporated and their impact on the performance for different query types was measured. Finally, we found that the mixed strategy which applies the most suitable ranking function to each query type yielded a significant increase in precision relatively to the baseline and to employing any examined strategy in isolation on the entire set of user queries.

Value:

The main contributions of this paper are: (i) the alternative approach for compact and concise representation of search results, which we implemented in the BoW on-line bibliographical catalog; and (ii) the unified or mixed strategy for search and result representation applying the most suitable ranking function to each query type, which produced superior results compared to different single-strategy-based approaches.

Keywords: hierarchical bibliographical catalog, hierarchical search, topic ranking, coordination level ranking, result representation.

Introduction

The need for concise display and user-oriented manipulation of retrieval results has been addressed by various systems (Berenci *et al.*, 1998). Among others, Bead (Chalmer and Chitsons, 1992) and LyberWorld (Hemmje *et al.*, 1994) depict clustering patterns in a document space using three-dimensional visualization schemes. TileBars (Hearst, 1995) displays the distribution of query terms within each document to locate its relevant parts. Ulysses shows a lattice of terms and documents that can be searched in various and integrated ways (Carpineto and Romano, 1996). Berenci *et al.* (1998) concentrate on all the possible subsets of query terms (i.e., sub-queries) that can be generated from the user query, showing their distribution in the set of retrieved documents and letting the user select the associated set of documents. Anderson *et al.* (2002) display the frequency with which query terms are found in a document using simple pie charts. Dushay (2004) suggests employing a two-dimensional scatter plot with zooming to display bibliographic entries (called "virtual spines"). On the other hand, the most popular information systems

and search engines like Medline (htpp://www.pubmed.gov) and Google (http://www.google.com) present a long set of document surrogates in a ranked order and sometimes provide an additional link to related documents or subjects.

The success of the above systems in helping users make faster and more accurate document judgments has been uneven, mostly due to the wide range of information needs and also the relative effectiveness of different visualization forms (Anderson *et al.*, 2002). One of the main reasons for user dissatisfaction with current retrieval interfaces is the lack of a concise representation of the summarized content of all retrieved documents.

We suggest that a better utilization of the data organization model is a main key to improvement of the results representation scheme. As can be observed, many modern data repositories are organized into a hierarchy of topics or subjects. Searching within a hierarchy has two independent uses. One is for retrieval of information. The other is for insertion of new data – essentially on-line indexing, where new items are added to the catalog and need to be linked to the most relevant locations in the hierarchy.

This paper attempts to formulate a model that captures these two tasks (hierarchical retrieval of existing items and insertion of new ones) into one unifying algorithmic scheme, which, as shown below, also leads to an improved methodology for result representation. The proposed model is logically divided into the following three fundamental components or phases which need to be investigated and optimized:

- Searching the Topic Hierarchy
- Visual Representation of the Results
- Topic Ranking

This paper is further organized as follows. After presenting the main research problems and contributions we describe the proposed model in the Developed Model section. This section includes a detailed description of the interface and internal structure of the system. Further, the new improved methodology for each of the three phases of the model above is presented. In the following sections the proposed methodology is evaluated, the obtained results are discussed, and finally, a few possible future research directions are depicted.

Problem Statement

An obvious and natural approach to organize a large catalog is to use a hierarchical structure, which typically reflects the logical structure of the data. At the same time, many prevalent search facilities usually ignore the underlying hierarchical structure when presenting search results. Instead, they rank the retrieved items according to some relevance or importance metric and present the user with a linear list of results, which is typically quite long. Because of the vast amounts of information on almost all topics, one cannot systematically go over the whole set of results, and therefore, must rely on the ordering of the results by the search engine. Hence, one of the biggest challenges for modern information retrieval systems is handling the tradeoff between generating an accurate and concise list of matching search results on one hand (leading to high retrieval precision), and making this list complete and informative on the other hand (obtaining high recall).

(Kules *et al*., 2008) provides a broad overview of the existing systems that try to organize the search results by facets and/or existing topic or subject hierarchies, typically employing some automatic clustering methods. For example, faceted categorization systems like Dyna-cat (Pratt, 1997) automatically produced facets for sets of search results and showed that users were 50% faster in fact finding tasks using Dyna-cat over typical ranked list keyword search interfaces. In SERVICE system the facets are automatically generated by applying fast-feature classifiers (Kules *et al.*, 2006) over the top 100 results of a Google query, which are organized into known possible categories drawn from the Open Directory Project (ODP) and a database of US Government web sites (http://www.lib.lsu.edu/gov/tree): Topic, Geography and US Government. Clicking on a category filters (or narrows) the displayed results to just the pages within that category. Moving the pointer over a category highlights the visible search results in that category in yellow. Moving the pointer over a result highlights all the categories in the overview that contain the result. The GRiDL prototype displays search result overviews in a matrix using two hierarchical categories (Terveen *et al*., 1999). The users can identify interesting results by cross-referencing the two dime. The TopicShop Explorer interfaces combine a hierarchical set of topics with a user-controlled detailed list of titles and attributes within each topic (Furnas & Rauch, 1998). The NCSU library catalog (Antelman *et al.,* 2006) provides

categorized overviews of search results using subject headings, format, and library location.

Searching in the hierarchy imposes additional challenges. Given the topics selected by the hierarchical searching process, the question is how to display them on the screen. The dilemma is how to reconcile two contradicting considerations: on one hand we would like to preserve topological locations of the topics in the hierarchy, but on the other hand we would like to sort them according to their relevance rank.

There are several examples of an attempt to deal with this problem. Examples of hierarchy-based: The Berkeley Cha-Cha search engine (Chen *et al.*, 1999) employs grouping based on topological proximity in the index hierarchy, and the clustering: Clusty search engine (http://www.clusty.com) provides a list of clusters of the output links, ranked by the number of links in each cluster, in parallel to the list of links. The Open Directory (http://search.dmoz.org/) and Yahoo! Directory (http://dir.yahoo.com/) accompany each search result with a path of where it appears in the directory's hierarchy.

All these are compromises which favor one consideration over the other. Our approach provides a solution to the above problem as described in the next sections.

The core of a hierarchical searching method is the ranking function that determines the selection of the most relevant topics at every level of the index (the third component of the proposed model). A variety of ranking functions have been proposed in the literature, the most popular being $tf \times idf$ (Salton and Buckley, 1988) and coordination level matching (CLM) (Van Rijsbergen, 1979). However, as has been pointed out (Fan *et al.*, 2004) using only one fixed ranking strategy for a variety of query types, user needs, and document collections might lead to serious performance problems.

Thus, an additional goal of this work is to systematically explore the relationship and impact of applying different ranking functions on various query types. Eventually, we expect that this study will help us achieve our final goal: to identify and develop a better unifying ranking strategy which would identify and apply the most suitable function to every user query type and information task.

Main Contributions

This paper presents an empirical study that aims to explore the user information needs and accordingly improve the search process in an online hierarchical bibliographic catalog.

First, we focus on the idea of finding a topic in a hierarchy as the result of a search, and ranking whole topics, rather than individual documents (or entries). Based on this principle, we propose a novel representation of the weights and locations of the retrieved results. Thus, for searching the hierarchy and result representation (the first two components of the model as listed in the Introduction) we take the following unified approach. We suggest that given a hierarchical structure, it is desirable for search procedures to point to relevant locations within this hierarchy, as a supplement to providing a flat and disconnected listing of individual results. For example, in the context of searching pictures, a query of "baby" may return pointers to a couple of albums predominantly filled with baby pictures, rather than just a mixed list of individual pictures from these and other albums. This approach provides the user with a wider context of related documents, within which the best data to answer the query can be found. Similarly, relevant locations for the insertion task can be found by simply using the new item to define a query and then utilizing the same technique for item insertion as for search. In either case, the most relevant locations in the hierarchy can be indicated by graphical cues that make them stand out from the general structure. For example, in BoW we use increased font size as illustrated below in Figures 2 and 3.

In order to implement the idea of retrieving the best matching topics from the topical hierarchy, we developed a specialized term weighting scheme suitable for use in a hierarchy. We further demonstrate the utility of combining multiple independent optimizations incorporated in the proposed weighting scheme (such as vocabularies based on 5-grams, special weights for headings, and special treatment of authors).

The last step in the search process is topic ranking by measuring the overlap between the given query keywords and the topic vocabulary. While a wide variety of ranking metrics is employed in the literature (cf. the subsection "Ranking Functions Examined" below), a systematic investigation of their effectiveness for different user query types and information needs is still required. Therefore, we conducted a user study

and examined several vector-space-based ranking functions including a classic *coordination level matching* (*CLM*) approach (Van Rijsbergen, 1979) as a baseline along with more complex weight-based methods from the literature.

Inspired by the analysis of the existing ranking methods, we then propose a new weight-based ranking function. Its underlying intuition is that the most relevant topics for a query are expected to contain many of the query terms at the top ranks of their keyword vectors. The top ranks are determined by the highest weights of keywords for a topic. We call this ranking strategy the *minimal term distribution gap (MTDG)*. Thus, the smaller the distribution gap between the terms is, the more relevant the topic is to the query.

The various weight-based functions employed in our experiments were constructed such that each of them reflected and tested the impact of an additional weighting factor on search performance. Our investigation of various ranking functions from the literature for different user queries and tasks shows, that different functions perform better for distinct query types and user needs. For example, we found that pure coordination level matching is more effective for very long queries (e.g. those created from new items for insertion) and for queries on authors, while the new *MTDG*-based search performs substantially better (by over 30%) for short (2-3 word) keyword queries.

Finally, inspired by the above observations we establish a unified ranking method, which employs the best fitting function (among the examined ones) for each query type according to our findings. The unified approach increases the topic retrieval precision by up to 50-90% over the other methods, when applied on the entire mixed set of user queries of various types.

Field Name	Content
Author(s)	Shailadh Nagar, Ajit Banerjee, Anand Sivasubramaniam, and Chita R. Das.
Title	Alternatives to Coscheduling a Network of Workstations.
Journal / Publisher	Journal of Parallel and Distributed Computing, vol. 59, pp. 302-327.
Publication Year	1999
Annotation	Compares 9 combinations of what to do when waiting for a message (spin, spin and yield, or spin and block) and what to do when it arrives (nothing, interrupt and reschedule, or periodic rescheduling). Evaluations using a mix of real applications indicate that periodic boost is the best.

Field Name	Content
Author(s)	John K. Ousterhout.
Title	Scheduling Techniques for Concurrent Systems.
Conference	In the 3rd International Conference on Distributed Computing Systems (ICDCS), pp. 22-30.
Publication Year	1982
Annotation	Examines scheduling policies for concurrent systems where the processes interact strongly. Two-phase blocking is suggested as an improvement for short-term scheduling, and coscheduling as a guideline for long-term scheduling.

Table 1: The content of a typical bibliographic entry in BoW. The upper table represents a journal article entry, while the lower one displays a conference paper entry.

The Developed Model

The BoW Bibliographical Catalog

To illustrate and evaluate our developments we used an on-line bibliographical catalog, called BoW, dedicated to the somewhat limited domain of parallel systems. BoW stands for "Bibliography on the Web". The goal of the BoW project (Feitelson, 2000) is to create a user-friendly working environment for the construction, use, and maintenance of an on-line bibliographical catalog.

The key idea is that this be a communal effort shared by all the users. Thus, every user can benefit from the input and experience of other users and can also make contributions. In fact, the system tabulates user activity, so merely searching through the catalog and exporting selected items already contributes to their ranking in terms of user interest. A prototype implementation is available at http://www.bow.cs.huji.ac.il. The entries in the BoW catalog are surrogates for scientific publications: journal papers, conference papers, and books. Each entry contains the publication's authors, title, publication details (journal or conference, volume, pages, date) and possibly a brief user annotation. Examples of typical bibliographic entries can be viewed in Table 1. Full text is not stored as part of the catalog, but external links are supported. The search and indexing procedure described below only uses the stored data, namely authors, title, and annotations. This provides enough data to work with while reducing the amount of data that needs to be handled (Kerner and Lindsley, 1969; Montejo-Raez *et al.*, 2005).

The heart of the BoW catalog is a deep (multi-level) hierarchical index spanning the whole domain. The nodes in the hierarchy are called *concept pages*. Pages near the top of the hierarchy represent broad concepts, while those near the bottom represent narrower concepts. The depth of the hierarchy should be sufficient so that the bottommost pages only contain a handful of tightly related entries (as opposed to Web directories such as Yahoo! and CORA (McCallum *et al.*, 2000) which are shallow relative to the number of documents they contain).

A concept page has a heading that defines the concept, and links to sub-concepts and to actual entries. This is analogous to a folder in a file system, which may contain sub-folders and actual files. The hierarchy of concept pages is constructed manually by the site editor based on a thorough knowledge of the topic domain. New concept pages can be added as this knowledge evolves. Entries can be linked to multiple concept pages, if they pertain to multiple concepts. Likewise, they can be linked at different levels of the hierarchy, depending on their breadth and generality.

A sub-tree containing all the concept pages and entries reachable from a certain (high level) concept page is referred to as a *topic*. The topic is identified with the concept page at its root; thus the topic heading is just the heading of this concept page. The size of a topic is the number of entries it contains. As explained below, our search procedure is based on associating a vector of keywords with each topic. The vocabulary used is based on topic headings, entry authors and titles, and user annotations, and is therefore uncontrolled by the system, so users can also query the system using natural language (Blair, 1990).[1]

Our prototype catalog on parallel systems contains about 3500 bibliographical entries. These entries are linked to about 140 concept pages, arranged in a hierarchy that has a typical depth of 4 or 5 (Figure 1). The catalog is navigated using a conventional browser. Normally three frames are available, with functionalities that are similar to those that are now common for file browsing on desktop systems. The first frame on the left in Figure 2[2] shows the hierarchical concept index. Initially it shows the list of top-level concept pages. Clicking on one of them expands that branch of the hierarchy by one level, and also makes this the *selected* concept page. Any entries that are linked to this concept page are listed in the second frame (on the right). Clicking on an entry from this list makes it the *current* entry. The third frame (on the bottom) displays the surrogate of this entry, including all the bibliographical data, user annotations, and additional links.

[1] Note that as opposed to the MeSH hierarchy of headings used as a controlled vocabulary of concepts for search in Medline (htpp://www.pubmed.gov), in our case the topics' headings are part of the catalog and clicking on one of them displays all the corresponding subtopics and entries that are attached to it in the hierarchy (rather than a list of the individual and disconnected matching entries).

[2] This figure actually shows the configuration for viewing search results, but the main frames and their layout are the same as for browsing.

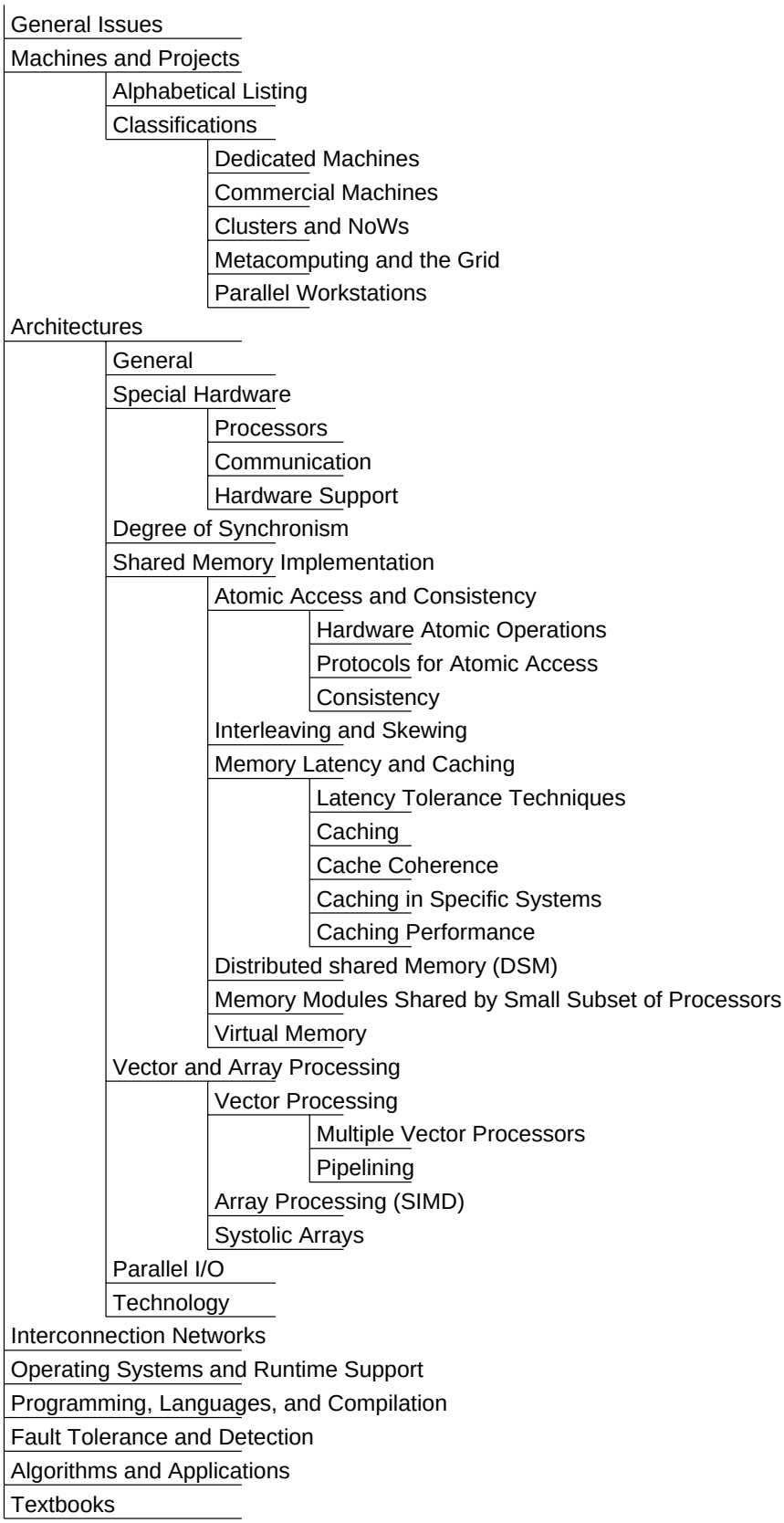

Figure 1: The BoW concept hierarchy showing some of the structure of two top-level concepts.

Available operations on the current entry include marking it for export, adding an annotation, and adding links. This includes links from additional concept pages to the entry, links between this entry and related entries (e.g., from a preliminary version of a paper to the final version), and links to external resources such as the full text.

In parallel to the hierarchy of concept pages, a hierarchical index of characterizing keyword vectors for each topic is constructed. This index has the same structure as the hierarchy of concept pages, and is in fact based on its contents. Each node in the index is a vector of keywords which represent the vocabulary of the corresponding topic in the hierarchy. Since each topic encompasses all the concept pages and entries in a sub-tree of the hierarchy, all these concept pages and entries should be taken into account when constructing its keyword vector. The keywords are selected automatically as the most relatively significant words for this topic, which also differentiate it from its sibling topics. The complete and detailed description of the keyword selection algorithm is presented in our previous work (Geffet and Feitelson, 2001) and is briefly summarized in the next paragraphs.

The vocabulary of a topic is based on all the concept page headers and entry contents in its sub-tree. First, all the words are stemmed and a stop list is applied to remove common English words that do not represent any specific topic. The remaining words are then replaced by all the five-grams of letters in them, shifting right letter by letter from the beginning to the end. For example, “algorithm” will be turned into “algor”, “lgori”, “gorit”, “orith”, and “rithm”. This has two desirable effects: first, related words have many five-grams in common, so using one word in a query will typically have a good match with other related words that appear in the catalog, and second, long words will tend to have a higher weight in the comparison process because they will be represented by more five-grams (Geffet and Feitelson, 2001). Words shorter than 5 letters are included in full. From now on the terms “five-gram” and “word” will be used interchangeably, except in cases where we need to consider the words that appear in the original query (in particular, for ranking function definition).

An additional advantage of BoW is that our data is semi-structured. In particular, it is easy to give special treatment to author names and topic headings. Author names are inserted into the vocabulary as is, without being parsed into five-grams, because in this

case we don't want related names to be identified with each other. We also constructed and employed an acronym thesaurus, since our data contains many names of projects, systems, and tools which are often referred to by acronyms.

Words from topic headings are given extra weight in the vocabulary. Weights are used when assessing the match between the vocabulary of a topic and that of a query. Normally, the weight of a word is the number of entries in the topic in which it appears. But since many queries are topical, it is important to ensure that the topics with query words in the heading will be assigned substantially higher weights than those including query words only in their entries contents. Thus, for a given word *w* and a topic *t*, the word's weight in the topic's vocabulary, $Voc_t(w)$, is calculated as follows:

$$Voc_t(w) = termfreq(w, t) + inheading(w, t) \cdot [A + B \cdot subtopics(t) + entries(t)]$$

where *termfreq* is the overall number of occurrences of the word *w* in the different entries of the topic *t*, *subtopics* is the number of subtopics of the topic *t*, *entries* is the number of entries that are included in the concept page of the topic *t*, and *inheading* is a binary predicate that evaluates to 1 if word *w* is in topic *t*'s heading. This then adds the terms in the square brackets to the weight, including constants *A* and *B* and two additional terms that reflect the topic's size. In our experiments the constant *A* was set to 100, which exceeds (typically by an order of magnitude) in our corpus any existing term frequency in a topic's content, and *B* to 5, which is slightly higher than an average number of direct subtopics. These two constants thus ensure that the weight of words appearing in the topic heading is always higher than the weights of words which appear only in the contents of some internal entries. The values of *A* and *B* might be updated accordingly for larger data collections. Note that in order to eliminate the scaling effect[3], the $termfreq(w,t)$ values were counted only once for an entry and divided by the number of entries in the topic. This normalization method yields uniformly distributed weights for topics of varying sizes (Geffet and Feitelson, 2001) and was also shown to be more effective in our case than the popular *tf*×*idf* metric (Salton and Buckley, 1988).

In summary, the idea of this sort of index is to construct a pure content-related (reflecting) language, while dropping out all the meaningless words. One may wonder

[3] The scaling effect problem (Korfhage, 1997) appears when the counter values in "small" topics are generally lower than in "big" topics, leading to an assignment of all the keywords to the bigger topics.

why not use the full text vocabulary of the entries for indexing purposes. However, previous work has shown that a significant increase in accuracy and a real decrease in computational cost can be achieved by reducing the size of the vectors (Koller and Sahami, 1997).

Phase I: The Unified Approach for Searching the Topic Hierarchy

The initial construction of the keyword index and its updates are executed off-line, repeated at regular intervals.

At the on-line phase, for each user's action (search or new entry insertion) a query is created and handled by the unified searching procedure. This is executed recursively on the concept index, starting with the top-level topics. In essence, the query vocabulary vector is matched against the keyword index vectors of the different topics in a vector-based manner (Geffet and Feitelson, 2001). The search then proceeds recursively from the top-level topics, and choosing the most suitable sub-topic(s) at each step.

This approach provides better accuracy than the traditional flat query-document matching schemes over a structured document corpus as shown in (Koller and Sahami, 1997) and (McCallum *et al.,* 1998). The main advantage of the hierarchical method is that at every stage the set of the sub-topics to be investigated next is pruned, and the decision to be made by the classification process is simplified and more focused.

Phase II: The Unified Approach for Visualization and Manipulation of Retrieval Results

Once the most relevant topics for a query are identified, the question is how to present this to the user. The solution we adopted in BoW is to use additional graphical modalities, and not just location. In particular, the same hierarchical structure of topics and sub-topics (as described in the previous sections) is utilized both for manual browsing and for displaying search results and suitable locations for insertion of new entries. Specifically, we use location on the page to represent the hierarchical structure of the concept index, and where the best matching concepts are located within this hierarchy, as shown in Figures 2 and 3. We use font size to represent importance and relevance to the query, as reflected by the score achieved by each topic.

One of the buttons in the toolbar is the *search* button, which initiates a search based on keywords and/or authors. The right-hand frame is used to display a list of matching entries, while the left-hand frame is used to indicate which topics are the most relevant for the query. Once identified and ranked, the relevant topics are displayed by opening the hierarchy until they are exposed, and emphasizing them by using a larger font; the larger the font is, the higher the relevance of the topic is to the query. By selecting one of the highlighted topics its sub-topics are opened in the hierarchy and its contents (sub-headings and entries) are displayed in the right-hand frame. In the case of author queries, the selected topics can be taken as a summary of the research areas in which the query author is active (Figure 2).

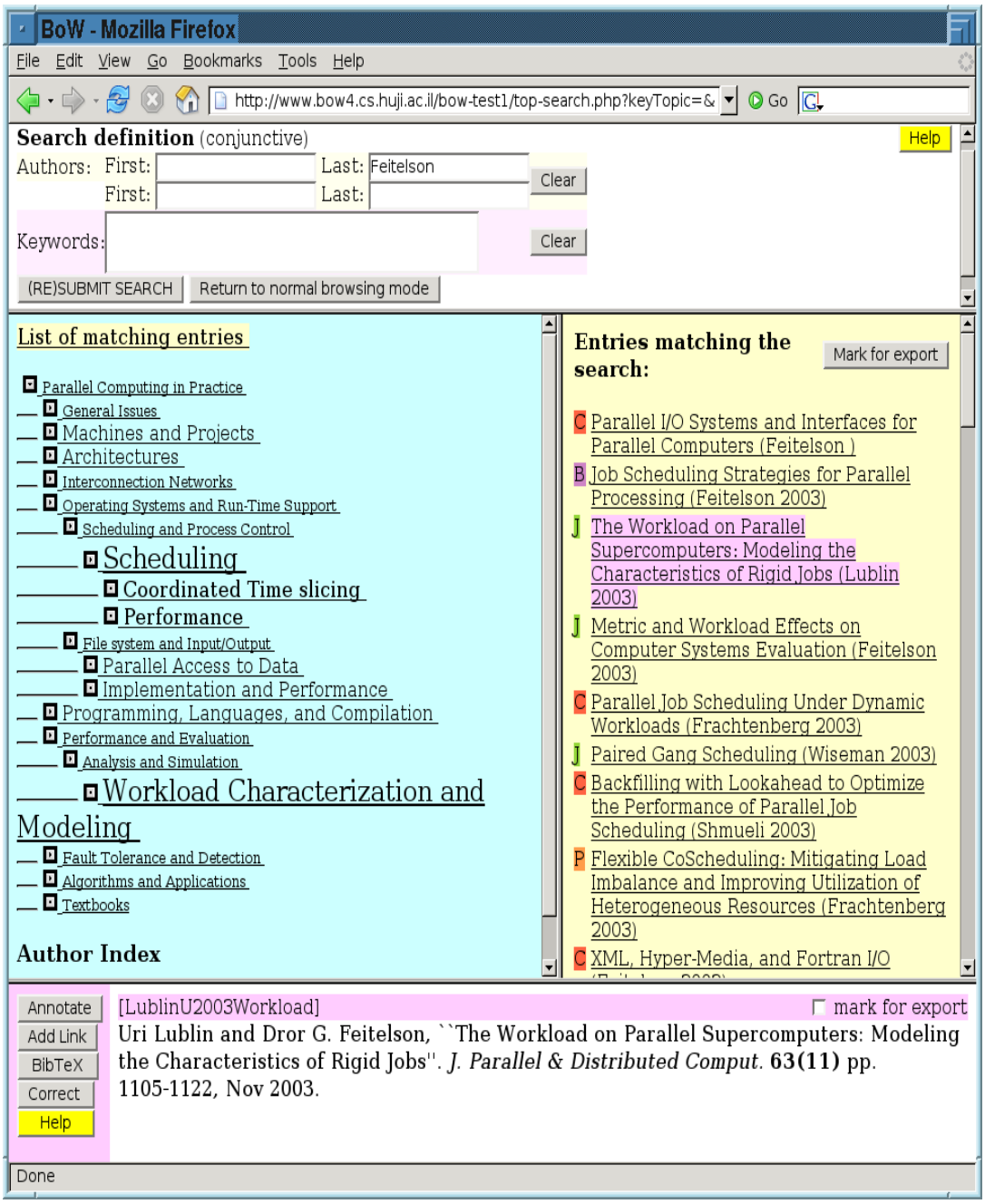

Figure 2: Display of the results of an author search. The large panel on the left shows the concept index. The opened and emphasized topics identify the query author's research areas. The right-hand panel provides a list of documents co-authored by the query author. Clicking on an entry shows its details in the bottom panel.

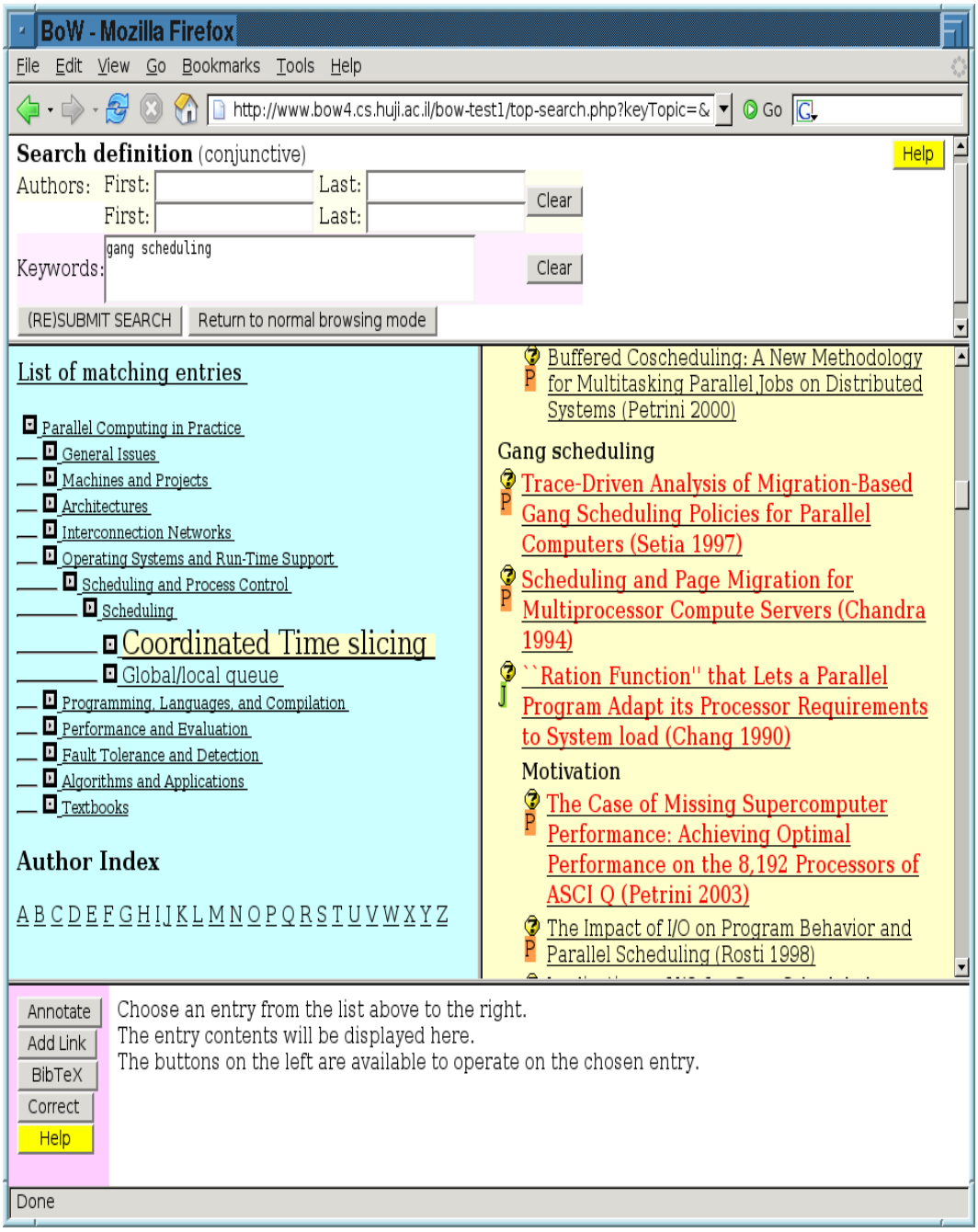

Figure 3: Display of results of a keyword search. In the concept index, the most relevant topics are opened and emphasized with a larger font. Clicking on one of them shows the entries it contains in the right-hand panel, with those that precisely match the query emphasized in red.

Examples of how this works out are shown in Figures 2 and 3 for author and keyword queries, respectively. Instead of showing the concept index fully opened along

one or more branches, as would happen during normal browsing, in a query response only branches leading to high-ranking concept pages are shown. For example, in the query on “gang scheduling”, the branch starting with “Operating Systems and Runtime Support”, continuing with “Scheduling and Process Control", and ending with “Scheduling” is shown, providing context for the highest ranking topic for this query “Coordinated Time Slicing”, and the second highest ranking “Global/Local Queues”. If we click on the “Coordinated Time Slicing” topic, its contents are displayed in the right-hand frame. As can be seen it contains a sub-topic “Gang Scheduling”, which particularly focuses on the subject of the given query. Thus, the searching procedure effectively retrieves all the relevant entries at once by identifying this topic. Importantly, this includes related entries that have been linked to this topic even though they do not contain the search term explicitly. Moreover, the results are grouped together and displayed within the context of related topics, which might be further explored by the user if needed.

This approach for hierarchical topic searching has several important advantages compared to the individual entries search. First, typically, the number of the most fitting topics returned by our unified topic searching procedure is an order of magnitude smaller than the number of matching entries retrieved by a regular query-entry matching search. Hence, the user effort required for relevance judgment of the output is significantly reduced. Second, all the entries located under some relevant topic are usually highly relevant as well (since they were linked to the topic by a human expert as described below), which in turn increases the system precision. This also avoids the problem, as typical for most information retrieval systems based on query-entry (or query-document) term matching, where a large fraction of the matching entries include the query terms as a passing reference but do not really focus on the query subject as a whole, yielding a lot of “noisy” irrelevant results in the output list. Furthermore, if a relevant entry uses a different terminology than the given query, it will never be found by the query-entry term matching procedure. This scenario is unlikely to happen in the topic search, where relevance of an individual entry is determined by relevance of the topic it belongs to, rather than by the specific entry’s terms. Likewise, the recall of the system is increased by providing the context of related broader topics in the hierarchy for further browsing.

Our searching approach also differs from the "search by subject" utility which is employed in many popular systems (such as Medline and VUBIS-based OPACS (http://www.library.geac.com/page/vubiseng_LIB.html)). In these systems when searching by subject, the query terms are matched solely to the terms in the headings of subjects, while in BoW each topic is represented by all the prominent terms of all subtopics and entries in its sub-tree. Consequently, the resulting topics are those that are mostly relevant to the query, since they include the largest amount of relevant material, even if the query terms do not appear as part of their headings.

While traditional library OPAC systems do not support clustering of the results or "more like this" option, Jacso (2007) argues that OPAC systems should take advantage of their high quality, accurately tagged, metadata-rich records for automatic clustering. Recently, some "next generation" OPAC systems, like NCSU (Antelman *et al.*, 2006) started to provide automatic clustering option of relevant documents by a few predefined facets or dimensions, like Topic, Author, and Format. Once users enter a search query, they can explore the result set by selecting values from these dimensions. Each dimension value also lists the number of results associated with it. They also display a list of corresponding Library of Congress Subject Headings, which can be used to refine the search. However, as opposed to BoW, which uses a quite deep and fine-grained topical hierarchy combined with relevance ranking of the resulting topics, the NCSU system displays only a few facets and a flat list of subject headings with no relevance ranking of these headings to a given query.

Another button in the regular toolbar opens the *add entry* menu, which offers a choice of entry types roughly based on the types available in BibTeX (Lamport, 1994). Each entry type in BoW has a customized form that allows the relevant data to be entered. Submitting this form has the side effect of performing a search based on the submitted data, i.e. the unified procedure for searching the topic hierarchy is activated in order to identify concept pages to which the entry may be linked. However, the actual linking is left to the discretion of the user. This is done by displaying the topics in the search results with check buttons next to them; selecting a topic by marking its check button indicates that a link should be created from this topic to the new entry.

The actual mapping to font sizes is done using different HTML font sizes. Given the set of scores of all topics returned from the query, the minimal and maximal scores are found. This range is then divided into 4 parts. The bottom 10% of the range are rendered with a font size of -1 (the top level of the hierarchy and branches that are only shown to provide context for lower topics are also rendered with this size). The range from the 10th percentile to the midpoint of the range is rendered with a font size of 0 (the default). The range from the midpoint to the 90th percentile is rendered with a font size of 1. The top 10% of the range is rendered with a font size of 2, provided the maximal score is higher than the midpoint by at least 2 points; otherwise, the range is too narrow and the maximal size is not used.

Phase III: Matching the Topic Ranking Methodology to User Needs

As was described in the previous section, the proposed unified procedure for finding the best matching topics to a given query was designed to handle both searching and insertion of new entries in a similar manner. However, the optimal topic ranking function to be used by this matching procedure might vary according to the type of the provided query (Fan et al., 2004). For example, queries for the insertion task include the whole content of the new entry. This can typically include a dozen words or even more (which are further split into multiple corresponding five-grams). But queries for search are much shorter. It has been observed that a typical web query contains only one to three words (Beitzel *et al.*, 2004).

In this context, we can also take advantage of the knowledge of the BoW catalog structure. Thus, another parameter to consider is the content of the query, e.g. queries including author names or other proper nouns might require a different treatment than queries which consist of common noun keywords. Some ranking functions may achieve higher recall while others are more precision-oriented. Thus, the choice of the ranking function also depends on the user's information needs and goals – getting a broad coverage of the existing relevant material on a subject or looking for a precise answer to a specific question.

Hence, a further investigation of the influence of the above factors on the ranking method performance is required. In this section we examine the behavior of a number of ranking functions that are designed to handle various query types and user information needs.

Ranking Functions Examined

Coordination Level Ranking

The basic approach, called *coordination level matching* (*CLM*), is to calculate the topic score by counting the overlap of query words, $QVoc$, with the topic's keyword vector, $TKeys_{topic}$:

$$CLM - score_{topic} = \left| QVoc \cap TKeys_{topic} \right|$$

A CLM procedure will always rank documents containing $n+1$ query terms above documents containing n query terms even if the top documents have little evidence for the presence of $n+1$ query terms and lower-ranked documents have a lot of evidence for the presence of n terms (Hiemstra, 1998). This strategy is supported by a user study (Wilkinson *et al.*, 1995) that found that people find it more acceptable to see documents that have many matching terms compared to one term matching many times.

In our previous work (Geffet and Feitelson, 2001) we applied the CLM ranking function to the task of inserting new entries. This was evaluated by 7-fold cross validation: in each experiment $1/7^{th}$ of the entries were removed, the vocabularies were constructed according to the remaining entries, and then the removed entries were used as queries. If the top-ranking topic matching each query was the one it originally came from, this was considered a "hit", and so on for lower levels. The average hit ratio for the top-level topics (with relatively large vocabularies) was quite high (89.2%-94.7%). Manually checking the entries that were misclassified revealed that in many cases they were ambiguous and had very short annotations (only one sentence). Hence, the search and insertion accuracy is influenced by the size of query vocabulary.

The major drawback of the CLM approach is that the score of a topic only reflects how many of the query words appear among the topic's keywords. However, this

information might not be sufficiently discriminative when handling short queries, which consequently leads to too noisy results. In truth, keywords are not all equal in the degree that they represent a topic: for example, a keyword that appears multiple times both in entries and in the topic heading should carry much more weight than a keyword that appears only once in a single entry. Hence, our expectation is that in this case weight-based ranking (rather than binary scoring) combined with the CLM approach may boost the performance.

Two main normalization strategies were suggested in the literature to handle the above problem of CLM. We present them below and further revise and explore their behavior in the next section.

The first and simplest approach is to adapt the w*eighted coordination level (WCL)* technique of Wilkinson et al. (1995) and sum the terms' five-gram weights (frequencies of appearance), $Voc_{topic}(f)$, in the topic weighted vocabulary vector, rather than incrementing the score by one point for each matched five-gram. This leads to the following formula for the score.

$$WCL - score_{topic} = \sum_{f \in QVoc} Voc_{topic}(f)$$

where five-grams that do not appear in the topic vocabulary are given zero weight. Note that, in particular, this will emphasize topics with keywords that appear in the topic heading, because of the artificially inflated counts of words that appear in the heading as described above.

However, this function might still suffer from the "non-coordination level" problem (Hiemstra, 1998) that is also common for *tf*×*idf*-based weighting: sometimes documents containing n query terms are ranked higher than documents containing $n+1$ query terms. On the hand, it was observed that weighting measures that are more like coordination level ranking perform better on the TREC collection, especially if short queries are used (Hiemstra, 1998). For example, Wilkinson et al. (1995) and Hiemstra (1998) show that the cosine, *tf*×*idf*, and CLM-like (e.g. Okapi) measures applied to short queries of up to 10 words behave differently, with the cosine and *tf*×*idf* measures performing far worse. They propose that it is advantageous to highly rank documents that contain all the query terms even if they are not highly ranked by the similarity measure.

Furthermore, Wilkinson et al. (1995) found that the CLM alone did not work well, while the WCL captured a larger amount of relevant documents, and the Okapi metric worked the best.

Thus, the best performance was obtained by some combination of CLM with term frequency weights. This approach was used in (Tan *et al.*, 2004) where a ranking function incorporates the CLM normalization factor by adding to document weight the number of query terms in the document divided by the total number of query terms.[4] Similarly, Mitsuhiro and Naohiko (1999) in their MEITSER system utilize a modified CLM scoring approach, where they added the above CLM factor to the *tf*×*idf* weights and show that this significantly improved the performance also for long queries.

We employ a slightly modified normalization scheme that adds a factor that ranks topics according to the number of query five-grams they contain. But in contrast with the simple binary criterion employed before, here the relative number of five-grams present is squared, to make this factor more discriminative and sensitive to every missing term. Hence, we define the *normalized weighted coordination level (NWCL)* function, as follows:

$$NWCL - score_{topic} = \left(\sum_{f \in QVoc} Voc_{topic}(f) \right) \left[\frac{\left| QVoc \cap TKeys_{topic} \right|}{\left| QVoc \right|} \right]^2$$

Ranking Topics by Minimal Term Distribution Gap

Further analysis of experimental results reveals that many of the retrieved topics gained a high overall score only thanks to five-gram(s) representing one word of the query that had a very high weight in that topic, while the other query words have a very low (or no score) for the topic. Usually, in such cases the topic is not very relevant to the query and so should not be returned at the result. For example, a query on "optical network" may

[4]Tan *et al.* (2004) and Cormack *et al.* (1998) use the *cover density* ranking metric by adding a factor that measures the distance between query terms in a document. However, this approach is effective in case of document ranking but is not applicable for ranking topics, as the keywords in topic vectors were selected according to their importance to the topic and no information on their distance is preserved in the vectors.

retrieve an irrelevant topic "Point-to-point networks" since the word "network" appears in the heading and therefore gains a high weight. Our expectation for a truly relevant topic is that it should include most if not all of the query words as keywords, preferably all with high weights.

Therefore, we propose a new variation of the topic ranking method based on frequencies of appearance (weights) of the query words in the topic termed as the *minimal term distribution gap (MTDG)*. This metric follows the rationale that all the query terms should be equally important and thus roughly evenly ranked at the top of the topic vector. The normalization factor determines that the terms are equally frequent in the topic and the frequency sum ensures that they are overall high ranked.

The problem is that both the query vocabulary *QVoc* and the topic vocabulary Voc_{topic} are expressed in five-grams. Since five-gram-based scoring tends to favor long keywords which produce more five-grams, we select the highest-weighted five-gram to represent each word in the query. Denoting five-grams derived from query word w by $g \in w$, we therefore define:

$$weight_{topic}(w) = \max_{g \in w} Voc_{topic}(g)$$

Using this, the weighted score for a topic will be:

$$MTDG - score_{topic} = (\sum_{f \in QVoc} Voc_{topic}(f)) \frac{\min_{w \in Q}\{weight_{topic}(w)\}}{\max_{w \in Q}\{weight_{topic}(w)\}}$$

where Q is the original query (in words, not five-grams). In particular, if any query words are totally missing from the topic the topic's score will be 0.

Finally, the last variation is just combining both of the above normalization factors (from NWCL and MTDG) into one formula. We term this method as the c*ombined MTDG&NWCL*:

$$COMBINED - score_{topic} = (\sum_{f \in QVoc} Voc_{topic}(f)) \frac{\min_{w \in Q}\{weight_{topic}(w)\}}{\max_{w \in Q}\{weight_{topic}(w)\}} \left[\frac{\left|QVoc \cap TKeys_{topic}\right|}{\left|QVoc\right|}\right]^2$$

Note that these weighting schemes mainly affect queries with two or more words. For one word queries all the above formulas reduce to the initial form of summing the term weights (WCL).

One-word queries	backfilling, deadlock, Ethernet, grid, kernel, middleware, robustness, paging, workload, router, testing, scalability, protocol
Long (2-3 word) queries	adaptive scheduling, cluster computing, parallel computing history, performance optimization, client-server, fortran compiler
Acronyms	LAN, LRU, DSM, MPI, SCSI, RP3
Queries with typos	gang sceduling, kernel treads, load balansing, memory letency, flow kontrol, usr interface
Authors	Bal E. Henri, Yang Yuanyuan, Reed Daniel, Van Steen Maarten, Patt Yale N., Bertossi Alan (A.), Mellor-Crummey John M.

Table 2: Examples of various query types used by the judges.

As pointed out by previous work (Wilkinsion et al., 1995; Fan et al., 2004), it seems that various types of queries (e.g. long vs. short queries) are influenced by different factors and best performance might be reached by different ranking methods. Hence, our goal is to investigate the behavior of the above functions in different cases and query types and identify an optimal method for each case. The methodology and results of this investigation are shown in the next section.

Evaluation of the Developed System

We have compared the performance of the proposed ranking functions in a manual evaluation experiment. Finding suitable human assessors for the system evaluation was quite difficult, partly due to the narrow professional domain of the BoW material. Finally, we managed to find two highly qualified judges, both experts in the field of parallel systems, who independently created and tested two sets of over 200 queries. Each judge's set comprised about 100 author names queries and 100 keyword queries on the various subjects covered by the BoW catalog (e.g. "network computing", "data compression", "parallel job scheduling").

Approximately 50% of the keyword queries consisted of two or three words, the rest were one-word queries. There were also a few queries with typos (5%–8%, which seems like a reasonable relative number of typos for a typical user) in each set and 10 acronyms (as is also quite typical for an average user). The acronyms were automatically

score	**R***	**P***
3	mostly relevant results	very few non-relevant results
2	sufficiently many relevant results	some non-relevant results
1	few relevant results	many irrelevant results

Table 3: Scores used by the judges to evaluate query responses.

interpreted by the system through the pre-computed thesaurus and converted to their full wording. Table 2 exemplifies the various query types used in the experiment.

The judges were guided to evaluate the query results (all the highlighted exposed topics in the hierarchy) for each of the four weight-based ranking metrics and the baseline coordination level matching approach defined in the previous subsections. Two types of grading criteria were required for each query result: relative precision and relative recall, which capture the overall user impression and satisfaction with the results.

Query Type	Algorithm Version	Judge I		Judge II	
		R*	P*	R*	P*
One-word queries	MTDG	2.28	**2.25**	**2.47**	**2.67**
	NWCL	2.27	**2.25**	**2.47**	**2.67**
	COMBINED	**2.29**	**2.25**	**2.47**	**2.67**
	Baseline-CLM	2.05	1.93	*2.53*	1.73
2-3-word queries	WCL	**1.95**	2.29	2.07	2.34
	MTDG	1.90	**2.50**	2.04	**2.54**
	NWCL	1.93	2.35	**2.10**	2.53
	COMBINED	1.86	2.43	1.97	2.54
	Baseline-CLM	*2.12*	1.62	*2.21*	1.46
Author queries	WCL	2.07	**2.71**	1.73	2.85
	MTDG	1.92	2.42	1.70	**2.86**
	NWCL	2.45	1.92	**2.53**	1.84
	COMBINED	**2.46**	1.93	2.51	1.84
	Baseline-CLM	2.32	2.42	2.23	2.23

Table 4: Experimental results for various query types and algorithm versions, for the two judges. The best results among the ranking functions are marked by bold and when the baseline results are higher than those of the weight-based versions, they are denoted by bold and italics.

1. R* – corresponding to the relative level of recall achieved for the query, i.e. how many relevant topics were retrieved. This is interpreted as being relative to what may be expected, based on an understanding of the domain and some knowledge of the concept hierarchy structure.
2. P* – corresponding to the relative precision of the response, i.e. how many irrelevant results ("noise") were also retrieved.[5]

Scores were given numerically on a scale of 1 to 3 as specified in Table 3. These evaluation criteria require less user effort and allow for a more flexible estimation of the method performance than assigning a binary score of "relevant" / "non-relevant" for each individual result.

[5] Note that the standard precision measure is defined as the fraction of the search results that are relevant for the query, and recall is the fraction of the relevant material that was retrieved out of all the existing relevant material in the collection.

Query Type	Method	Judge I		Judge II	
		R*	P*	R*	P*
Author	Baseline:MTDG	0.157	0.103	0.487	0.305
	MTDG:NWCL	0.370	0.494	0.363	0.204
	NWCL:COMBINED	**0.991**	**0.995**	**0.952**	**1.000**
	COMBINED:MTDG	0.351	0.500	0.386	0.204
	Baseline:NWCL	0.163	0.157	0.597	0.638
	Baseline:COMBINED	0.139	0.171	0.550	0.638
	Baseline:WCL	0.199	0.215	0.479	0.274
	MTDG:WCL	0.685	0.284	0.903	0.969
	NWCL:WCL	0.397	0.338	0.362	0.165
	COMBINED:WCL	0.389	0.327	0.296	0.165
One-word keyword queries	Baseline:MTDG	0.550	-0.003	0.698	0.213
	MTDG:NWCL	**0.985**	**0.991**	**1.000**	**1.000**
	NWCL:COMBINED	**0.951**	**0.976**	**1.000**	**1.000**
	COMBINED:MTDG	**0.943**	**0.970**	**1.000**	**1.000**
	Baseline:NWCL	0.559	-0.036	0.698	0.213
	Baseline:COMBINED	0.499	0.014	0.698	0.213
All keyword queries	Baseline:MTDG	0.382	0.027	0.449	0.179
	MTDG:NWCL	0.824	0.843	0.639	0.475
	NWCL:COMBINED	0.831	0.838	0.718	0.597
	COMBINED:MTDG	**0.912**	**0.937**	**0.930**	**0.872**
	Baseline:NWCL	0.312	0.095	0.369	0.243
	Baseline:COMBINED	0.373	0.086	0.394	0.157
	Baseline:WCL	0.155	0.342	0.289	0.001
	MTDG:WCL	0.391	0.518	0.448	0.339
	NWCL: WCL	0.606	0.754	0.867	0.647
	COMBINED:WCL	0.386	0.571	0.548	0.442

Table 5: The linear correlation (Pearson coefficients) between the judgment values of various methods for each of the two judges. The methods with highest correlation values for both judges are marked with bold; this indicates that the methods exhibit similar behavior.

It is important to emphasize that the judges were not aware of the differences between the evaluated methods and had no knowledge about which one of them was the baseline and which were the weight-based ones. The final product of the evaluation experiment for each judge consisted of a table of approximately 200 queries across 5 searching methods with two grades for every query under each method.

Discussion

In order to analyze the obtained results, we calculated the average grades for each criterion, as graded by each judge, over different sets of queries. The full results are displayed in Table 4. For each criterion and query type, the top graded method is indicated by boldface. Note that, in some cases, the baseline is better than the weight-based methods, but only for criterion R* (relative recall). Since the query sets were distinct for each judge we could only measure their agreement by their grades' average values rather than by direct grade correlation per query. On the other hand, we did measure linear correlation for queries of the same set for different pairs of algorithms, as shown in Table 5. As the various ranking methods may behave differently for queries of certain types (presented in Table 2) we also computed the corresponding figures for each query type separately.

As expected, the correlation coefficient figures between different versions' results show an almost complete correlation for both judges between the MTDG, NWCL, and COMBINED methods for one-word queries. This is since the influence of the normalization factors only applies in case of longer queries. Hence, we did not ask the judges to test one-word queries for the WCL version. Another special category constitutes the acronyms. Similarly to the one-word queries they achieve almost identical average grades for all the weight-based methods, since they always appear in the topic vector either in the full form (including all the words) or in the acronym form. For author queries there is a very high correlation between NWCL and COMBINED (i.e. NWCL is the dominant factor in COMBINED), and for keywords – between MTDG and COMBINED, for both judges. This could be explained by the fact that for authors the number of query words that appear in the topic is a more crucial factor. Thus, to recognize topics relevant to an author with a high accuracy the system should require that both the first name and the surname appear in the topic, otherwise, the partially matching name may refer to a different person. As for keyword queries, in many cases all the query words would occur in the inspected topic, so the relative word frequencies play the role of the most discriminative factor. In addition, we notice that all the weight-based methods do not significantly correlate with the baseline.

While for one-word and acronym queries the weight-based methods yield significantly higher grades for both judgment criteria over the baseline, for longer queries the weight-based scores achieve a somewhat lower R*, but a much higher P* compared to the baseline ranking. Both judges consistently evaluate our proposed MTDG metric as the one with the highest P* grades for all query types except for authors.

We also observe that typos have little influence on the results for any ranking algorithm including the baseline, since the favorable behavior of the system for typos is determined by using n-grams (rather than whole words) and it does not depend on any other parameters of the topic weighting strategy. Adding the acronym queries to the 2-3-word queries pool leads to quite similar results as well.

For authors, on the other hand, the top grades were produced by the baseline ranking, while both judges consent that the WCL strategy is the best of the weight-based methods, and its grades are comparable with the baseline performance. This fine behavior of the CLM ranking could be explained by the different nature of the author queries, which, as opposed to the regular keyword queries, are rather precisely specified and are less ambiguous, since there are few authors with identical first names and surnames. In addition, once an author name appears in some entry of the topic it is automatically treated as a keyword by our indexing procedure and thus cannot be missed or filtered by the competition as may happen to entry content words (as described in the section on the unified hierarchy searching procedure). Therefore, even the CLM ranking algorithm, which typically suffers from too broad and noisy results, is suited to handle such focused queries quite well.

	Judge I					Judge II				
Query Type	Best Method	R%	P%	F1	IMP %	Best Method	R%	P%	F1	IMP %
One-word queries	**W***	64.5	62.2	0.63	28.4	**W***	73.3	83.2	0.77	57.4
	Base	52.5	46.5	0.49		Base	76.6	36.6	0.49	
2-3-word queries	**MTDG**	45.0	75.1	0.56	41.1	**NWCL**	55.0	76.3	0.63	93.1
	Base	56.1	30.9	0.39		Base	60.7	22.8	0.33	
Author queries	**WCL**	53.7	85.3	0.65	-3.7	**WCL**	36.7	92.2	0.52	-14.8
	Base	66.0	70.9	0.68		Base	61.6	61.6	0.61	

Table 6: Recall and precision (in %) and F1 for the best performing functions for each case from Table 4 vs. baseline (CLM). F1 is calculated by the standard IR formula as a harmonic mean of recall and precision. The improvement over the baseline F1 is presented in column "IMP %". Note that for one-word queries all the weight-based methods produced very similar results which allows us to use any of them as the best method (denoted by W*).

The overall improvement rates in terms of recall and precision are summarized in Table 6. The table presents the best algorithm performance for each query type and judge and the improvements over the baseline. The metric used to determine the best method is F1, the harmonic mean of precision and recall. We use the relative R* and P* as approximations for recall and precision. In order to compute them as a percentage, all the grades were mapped into a scale of [0..1] simply by subtracting 1 and then dividing by 2 (because the original scale was [1..3]).

Overall, the weight-based ranking method results show a substantial increase in precision (by up to 55 percentage points), reaching 68–78% precision for keywords (compared to 27–39% for baseline) and 85–92% precision for authors (compared to 61–71% for baseline), with relatively little loss in recall (up to 19 percentage points). As shown in the table for 2-3-word queries the precision is two to four times higher with the weight-based methods. This consequently leads to improved F1 values for keyword

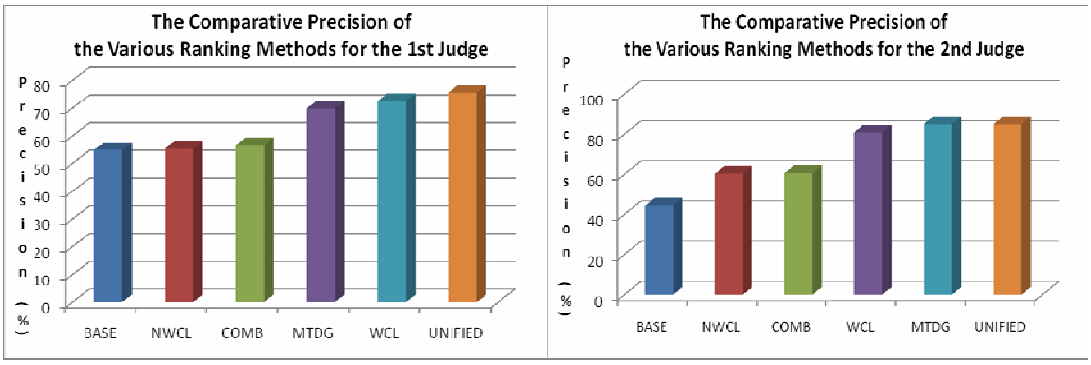

Figure 4: The comparative precision figures for the various ranking methods for each of the judges ranked in the ascending order. The unified approach (the right-most bar on the charts) has achieved the best precision values for both judges.

queries (according to both judges' average grades). For authors the weight-based methods performance is slightly worse than the baseline.

Remarkably, our experiment shows quite compatible results for both judges in a variety of cases and aspects. Specifically,

1. The weight-based methods always improve precision over the CLM matching baseline.
2. For all the keyword queries the proposed MTDG function yields the highest precision while NWCL achieves the highest recall.
3. For 2-3-word queries (without typos) MTDG produces the best F1 score.
4. The COMBINED and WCL methods typically exhibit weaker results than MTDG and NWCL. The possible reasons are that the COMBINED metric is too restrictive since it combines both normalization factors, leading to some decrease in recall, while the WCL version, which sometimes achieves quite good recall, is too permissive, since it uses no normalization constraints, which affects the precision.
5. For author queries the best weighting version is WCL producing results comparable to the baseline.

The CLM ranking (the baseline) usually produces higher recall scores than all the weight-based versions, as it generally retrieves larger resulting lists of topics. However,

this consequently significantly hurts precision with exception for very long queries, e.g. queries constructed from entries.

We conclude that the main contribution of the weight-based approaches is to improving the search precision, and the best metrics in this regard are MTDG and NWCL for keywords and WCL for authors. Inspired by these observations we can now establish a unified method, which employs the best fitting ranking function for each query type according to our findings above (Table 6).[6] To simulate the unified approach results we applied NWCL on the one-word queries, MTDG on the longer (2-3 word) keyword queries and WCL on the author queries. Then to quantitatively estimate the performance of the unified approach we computed the overall precision values produced by the examined ranking methods for all 200 queries for each of the judges and compared the results to the simulation of the unified method. As shown in Figure 4, the unified strategy yielded a significant increase in precision relatively to the other methods. The baseline WCL function obtained the lowest precision values (by 20-40 points lower than the unified method), while the performance of the WCL and MTDG metrics was only up to 5 precision points lower than of the unified method. For the Judge II our proposed MTDG method even produced similar precision to the unified method (85%), but with a lower recall, thus yielding a lower F1.

Thus, the retrieval system might either automatically employ the unified approach as we did in our simulation above, or give the users an option to choose the most suitable method for their needs, as follows:

- In case of a precision oriented search — MTDG / WCL will be selected for keyword / author queries, respectively,
- If recall is more important, but precision should be quite reasonable as well, NWCL might be the best choice for keyword queries,
- When a high recall is the user's only concern, the system will apply the CLM ranking procedure.

Finally, the BoW system is an experimental environment that implements most of the desirable features that are required for a 21st century OPAC as recommended in the literature (Young and Yu, 2004; Antelman *et al.*, 2006; Hilrith, 1995), such as natural

[6] We adopted this idea from the anonymous reviewer of the paper.

language search, relevance ranking of the results, more compact result representation (as users do not look at more than one page of the results), and "more like this" suggestions option (the most relevant sub-topics are retrieved and displayed rather than individual entries). Note, that most traditional library OPACs do not support the above features (Young and Yu, 2004; Hilrith, 1995). Furthermore, our ranking metrics and algorithms are based solely on term occurrence statistics and on the basic structure of the repository (fields like author, title, annotation), which do not depend on the specific domain or data sample of BoW. Therefore, we suggest that the same principles might be successfully applied to enhance the existing large scale OPAC systems.

Conclusions and Future Work

Information retrieval is typically concerned with the retrieval of documents out of a corpus that are relevant to a given query. The response to the user can be presented at various levels, ranging from a document reference number through a document surrogate to the full text (Korfhage, 1997).

BoW organizes its data in a deep and fine-grained hierarchy of topics and subtopics, and returns whole subtopics from this hierarchy in response to queries rather than long lists of individual matching documents. Our result visualization approach combines both exposing the most matching subtopics displayed within full topic hierarchy and emphasizing their importance and relevance with varying font sizes. This approach significantly reduces the user effort by concisely displaying the search output and provides the user with a wider context of related documents, within which the best data to answer the query can be found.

The system supports two main functionalities: insertion of new entries, and retrieval of existing ones. Interestingly, while a similar topic retrieval scheme was shown to be suitable for various application goals, we found that different ranking methods were best for different types of queries and information needs. This finding lends evidence to the notion that multiple approaches applied to the complexities of information retrieval behavior are needed. This is the key contribution of this research.

In particular, very specific and well defined queries like long entry-data-based queries and author queries were found to work well with coordination level ranking, i.e.,

by just counting how many query terms are matched. This approach also appeared to yield the highest recall. But for other keyword queries it was found to be better to combine the sum of keyword weight in each topic with a factor that measures whether most or all of the keywords are indeed present and evenly highly weighted.

Such a new strategy named, *minimal term distribution gap* (MTDG), achieves a much higher precision (increase of 30–50% over the CLM baseline) and F1 (increase of 34–78%). This implies that in order to obtain the best results, the search procedure should use different weighting schemes for different types of queries and retrieval tasks. Inspired by these observations we establish a unified method, which employs the best fitting ranking function for each query type according to our findings. This method improves the search performance by up to 50-90% for the mixed set of user queries.

Our algorithm was tested on a parallel systems bibliography with its specific structure, subject scope, and other characteristics. Future work may include testing the procedure on other data sets in different domains, to see how well it generalizes and what new issues are raised.

References

Anderson, T., Hussam, A., Plummer, B., and Jacobs, N. (2002), "Pie charts for visualizing query term frequency in search results", *Proceedings of Digital Libraries: People, Knowledge, and Technology: 5th International Conference on Asian Digital Libraries*, Singapore.

Antelman K., Lynema E., Pace A. K. (2006), "Toward a 21st Century Library Catalog", *Information Technology and Libraries*, Vol. 25, No. 3, pp. 128-139.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., and Frieder, O. (2004), "Hourly analysis of a very large topically categorized web query log", *Proceedings of the 27th SIGIR Conference on Research and Development in Information Retrieval*, pp. 321–328.

Berenci, E., Carpineto, C., and Giannini, V. (1998), "Improving the effectiveness of WEB search engines using selectable views of retrieval results", *J. UCS: Journal of Universal Computer Science*.

Blair, David. C. (1990), *Language and representation in information retrieval*. N.Y.: Elsevier.

Carpineto, C., and Romano, G. (1996), “Information retrieval through hybrid navigation of lattice representations”. *International Journal of Human-Computer Studies*, Vol. 45, pp. 553-578.

Chalmers, M., and Chitson, P. (1992), "Bead: explorations in information visualization", *Proceedings of SIGIR'92*. Copenhagen, Denmark, pp. 330-337.

Chen, M., Hearst, M. A., Hong, J., and Lin, J. (1999). “Cha-Cha: A System for Organizing Intranet Search Results”, *Proceedings of the USENIX Symposium on Internet Technologies and Systems*.

Cormack, G. V., Clarke, C. L. A., Palmer, C. R., and To, S. S. L. (1998), “Passage-based refinement (multi-text experiments for TREC-6)”, In Voorhees, E.M., and Harman, D.K. (Eds.), *Proceedings of the Sixth Text Retrieval Conference (TREC-6),* NIST Special Publication. Washington, DC: U.S. Government Printing Office.

Dushay, Naomi. (2004), “Visualizing bibliographic metadata: a virtual (book) spine viewer”, *D-Lib Magazine,* Vol. 10, No. 10.

Fan, W., Gordon, M. D., and Pathak, P. (2004), “Discovery of context-specific ranking functions for effective information retrieval using genetic programming”, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, No. 4, pp. 523-527.

Feitelson, D. G. (2000), “Cooperative indexing, classification and evaluation in BoW”, *Proceedings of the 7th IFCIS International Conference on Cooperative Information Systems*. Etzion, O. and P. Scheuermann (Eds.), 1901, pp. 66–77, Springer-Verlag, LNCS.

Furnas, G. W. and Rauch, S. J. (1998), “Considerations for information environments and the NaviQue workspace”, *Proceedings of the Third ACM Conference on Digital libraries, Pittsburgh, PA*. New York, ACM Press.

Geffet, M. and Feitelson, D. G. (2001), “Hierarchical indexing and document matching in BoW.”, *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, pp. 259–267.

Hearst, M. (1995), "TileBars: Visualization of term distribution information in full text information access", *Proceedings of CHI'95*, Denver, Colorado, USA, pp. 59-66.

Hemmje, M., Kunkel, C., and Willet, A. (1994), "LyberWorld - A visualization user interface supporting full text retrieval", *Proceedings of SIGIR'94*. Dublin, Ireland, pp. 249-259.

Hiemstra, D. (1998), “A linguistically motivated probabilistic model of information retrieval”, In C. Nicolaou and C. Stephanidis (Eds.), *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pp. 569-584.

Hildreth, Charles R. (1997), "The Use and Understanding of Keyword Searching in a University Online Catalog", *Information Technology and Libraries,* Vol. 16, No. 6.

Jacso, Peter. (2007), **"**Clustering Search Results -- Part II. Search Engines for Highly Structured Databases", *Online Information Review*, Vol. 31, No. 2, pp. 234-241.

Jacso, Peter. (2007), "Clustering Search Results -- Part III. The Synergy of Metasearching and Clustering", *Online Information Review*, Vol. 31, No. 3, pp. 376-382.

Kerner, C. J. and Lindsley T. F. (1969), "The value of abstracts in normal text searching", *Proceedings of the 6th Annual National Colloquium on Information Retrieval*. Philadelphia, pp. 437–440.

Koller, D. and Sahami, M. (1997), "Hierarchically classifying documents using very few words", *Proceedings of the 14th International Conference on Machine Learning*. Nashville, Tennessee, pp. 170–178.

Korfhage, R. R. (1997), *Information Storage and Retrieval*. N.Y.: John Wiley and Sons.

Kules, B., Kustanowitz, J., and Shneiderman, B. (2006), "Categorizing web search results into meaningful and stable categories using Fast-Feature techniques", *Proceedings of the Sixth ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC.* New York, ACM Press.

Kules, B. and Shneiderman, B. (2008), "Users can change their web search tactics: Design guidelines for categorized overviews", *Information Processing & Management*, Vol. 44, No. 2, pp. 463-484.

Lamport, L. (1994), LaTeX, a document preparation system. Addison Wesley, 2nd edition.

McCallum, A., Nigam, K., Rennie, J., and Seymore, K. (2000), "Automating the construction of Internet portals with machine learning", *Information Retrieval,* Vol. 3, No. 2, 127–163.

McCallum, A., Rosenfeld, R., Mitchell, T., and Ng, A. Y. (1998), "Improving text classification by shrinkage in a hierarchy of classes", *Proceedings of the 15th International Conference on Machine Learning*, pp. 359–367.

Mitsuhiro, S. and Naohiko, N. (1999), "NTCIR experiments at Matsushita: monolingual and cross-lingual IR tasks", *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pp. 61-71.

Montejo-Raez, A., Alfonso, L., and Steinberger, R. (2005), "Text categorization using bibliographic records: beyond document content", *Procesamiento del Lenguaje Natrural*, Vol. 35, No. 1135.

Pratt, W. (1997), "Dynamic organization of search results using the UMLS", *American Medical Informatics Association Fall Symposium,* Vol. 480, No. 4.

Rose, D., and Stevens, C. (1996), "V-Twin: A lightweight engine for interactive use", In Voorhees, E.M., and Harman, D.K. (Eds.), *Proceedings of TREC-5*. NIST Special Publication.

Salton, G. and Buckley, C. (1988), "Term-weighting approaches in automatic text retrieval", *Information Processing & Management*, Vol. 24, No. 5, pp. 513-523.

Tan, W., Chen, Q., and Ma, S. (2004), "THUIR at TREC 2004: QA", In Voorhees, E.M. (Ed.), *Proceedings of the 13th Text Retrieval Conference (TREC-13)*, pp. 575-580. NIST Special Publication.

Terveen, L., Hill, W., and Amento, B. (1999), "Constructing, organizing, and visualizing collections of topically related Web resources", *ACM Transactions on Computer-Human Interaction,* Vol. 6, pp. 67-94.

Van Rijsbergen, C. (1979), *Information Retrieval* (2nd edition). Butterworths, London.

Wilkinson, R., Zobel, J., and Sacks-Davis, R. (1995), "Similarity measures for short queries". In Harman D. K. (Ed.), *Proceedings of the 4th Text Retrieval Conference (TREC-4),* pp. 277-285. NIST Special Publication.

Yu, Holly and Margo Young. (2004), "The impact of Web search engines on subject searching in OPAC", *Information Technology and Libraries,* Vol. 23, No. 4.