2009-07-01

A Sophisticated Library Search Strategy Using Folksonomies and Similarity Matching

William Lund
billlund@icloud.com

Yiu-Kai D. Ng
ng@cs.byu.edu

Maria Soledad Pera

Original Publication Citation

Maria Soledad Pera, William Lund, and Yiu-Kai Ng. "A Sophisticated Library Search Strategy Using Folksonomies and Similarity Matches", Journal of the American Society for Information Science and Technology (JASIST), Volume 6, Issue 7, pp. 1392-146, July 29.

A Sophisticated Library Search Strategy Using Folksonomies and Similarity Matching

Maria Soledad Pera[1] William Lund[2] Yiu-Kai Ng[1*]

[1]3361 TMCB, Computer Science Department

[2]2060 Harold B. Lee Library

Brigham Young University

Provo, UT 84602, U.S.A.

Phone: (801) 422-2835

FAX: (801) 422-0169

Email: {mpera@cs.byu.edu, bill_lund@byu.edu, ng@cs.byu.edu}

ABSTRACT

Libraries, private and public, offer valuable resources to library patrons. As of today the only way to locate information archived exclusively in libraries is through their catalogs. Library patrons, however, often find it difficult to formulate a proper query, which requires using specific keywords assigned to different fields of desired library catalog records, to obtain relevant results. These improperly formulated queries often yield irrelevant results or no results at all. This negative experience in dealing with existing library systems turn library patrons away from library catalogs; instead, they rely on Web search engines to perform their searches first and upon obtaining the initial information (such as titles, subject headings, or authors) on the desired library materials, they query library catalogs. This searching strategy is an evidence of failure of today's library systems. In solving this problem, we propose an enhanced library system, which allows *partial, similarity matching* of (i) *tags* defined by ordinary users at a folksonomy site that describe the content of books and (ii) *unrestricted keywords* specified by an ordinary library patron in a query to search for relevant library catalog records. The proposed library system allows patrons posting a query *Q* using commonly-used words and ranks the retrieved results according to their

degrees of resemblance with *Q* while maintaining the query processing time comparable with the one achieved by current library search engines.

1. INTRODUCTION

Libraries, private and public, provide valuable sources of information, from century-old to the latest publications, which include journals, newspapers, textbooks, (non-)fiction books in many different languages, maps, audio and video scripts, etc. Library patrons from different age groups and educational background with diversity of information needs turn to libraries to locate information through library catalogs[1]. The library catalog has been a place to start searching for information, ranging from the old-fashioned card catalogs to the digital version used nowadays, since the catalog contains essential data (such as title, authors, or subject headings) of each library resource, e.g., books, maps, periodicals, etc. It is imperative to know that materials that are archived exclusively at libraries are not accessible by simply browsing or querying Web search engines, and the only alternative is to consult online library catalogs. Each library catalog, however, is defined by controlled vocabularies rather than commonly-used words, which are unintuitive to use by ordinary library patrons [Rethlefsen 2007]. As stated in [Borgman 1996], library catalogs are not designed for incorporating understanding on patrons' search behavior. What is more, Borgman [Borgman 1996] indicates that one of the main concerns regarding library catalogs is the fact that library catalogs must serve a heterogeneous population in terms of age, language, culture, subject knowledge, and computing expertise, and most of them are perpetual novices at information retrieval. As a result, library patrons who expect to locate the needed materials available in a library through its library catalog often encounter discouraging results, a tread that persists during the past few decades. Hence, the problem library patrons must deal with these days is not the lack of resources at libraries, but the

[1] The Library of Congress (http://catalog.loc.gov/help/contents.htm) defines the library catalog as a database of records that describe the collection of materials held by a library.

inflexibility in locating them through library catalogs. Since learning to use library catalogs is a tedious and time-consuming process, library patrons who demand easy-to-formulated queries and relevant information to be retrieved in a timely manner look elsewhere to satisfy their information needs. It is a common practice that library patrons first utilize Web searching tools, such as Google Scholar (http://scholar.google.com) or Amazon (http://www.amazon.com), in locating the primitive information (such as the title of a book or its authors) before querying library catalogs [Herrera 2007][2]. These user's behavior and expectations are influenced by their Web search experiences [Li 2008].

The ineffectiveness of online library catalogs for information retrieval has been the focal point of criticism [Larson 1991]. Even after almost two decades, the major design faults of online library systems still exist. These design problems include (i) the lack of user's understanding about the LCSH[3], (ii) difficulty in properly formulating queries against the library catalogs (as discussed earlier) compared with using simple keyword queries, (iii) information overload, i.e., searches that return too many results, (iv) search failures, i.e., searches that return no results at all, which according to [Yu 2004] are accountable for 10% to 40% of the searches performed on online library catalogs, and (v) irrelevant searches, i.e., searches that return library records that do not match the user's needs. Along with these problems, the lack of relevance ranking on retrieved results on many library search engines is another major concern of library patrons [Novotny 2004]. Even though ironically solutions to problems (iii)-(v) have been integrated into the design of current Web search engines, such as Google (http://www.google.com) and Yahoo (http://www.yahoo.com), none of the problems (i)-(v) has been addressed for improving library catalog searches [Yu 2004].

---

[2] De Rosa et al. [De Rosa 2005] show that 89% of library patrons start their searches using Web search engines and then query library catalogs.

[3] LCSH (Library of Congress Subject Heading) are words or phrases standardized by the US Library of Congress that provide a general concept (description) about library resources [Inouye 2001]. According to [Larson 1992], between 30% to 50% of the queries formulated against library catalogs, which are created using LCSH, retrieved no results.

Realizing all of these design faults, we have developed a significantly enhanced library system, called *EnLibS*, which handles all of the design problems listed above. *EnLibS* allows novice, as well as expert, users to quickly search for desired information without requiring special skills and advanced training in using library catalogs. *EnLibS* adapts Information Retrieval (IR) techniques, such as the Fuzzy Set IR Model [Baeza-Yates 1999], which extends the traditional Boolean IR models to establish the degree of similarity among (key)words in a query and in a document (i.e., library catalog record in our work). Furthermore, in processing a library query *EnLibS* consults "folksonomies," which have been used in Flickr (http://www.flickr.com), YouTube (http://www.youtube.com), and Del.icio.us (http://del.icio.us), to describe and identify pictures, videos, and Web sites, respectively. Folksonomies, which are also known as *social classification* or *social tagging* [Neal 2007], consist of free-text keywords compiled by ordinary users which describe different items, e.g., pictures, books, or videos. These keywords, which are not selected from a controlled vocabulary nor a pre-defined taxonomy, are determined by ordinary users, i.e., people who use Flickr, YouTube, LibraryThing, etc.

The design goals of *EnLibS* include (i) reducing the high percentage of *failed* library catalog searches, (ii) ranking the retrieved results according to their *degrees of relevance* to the corresponding query, rather than simply ordering the results by the date of publication, date a particular record was added to a library catalog, or the order in which they are retrieved, and (iii) maintaining the *query processing time* comparable with the processing speed of current library search engines.

In order to determine the efficiency of our enhanced library system, we have conducted two different performance evaluations. We first evaluated the effectiveness of *EnLibS* in reducing the number of failed searches using the Harold B. Lee Library (HBLL) transaction logs (from July 2006 to January 2007) and determined the percentage of zero-hits searches in the logs as compared with the ones generated by *EnLibS*. Second, we analyzed the overall performance of *EnLibS* in terms of (i) the degree of accuracy of the retrieved results

using the HBLL transaction log and (ii) the query processing time required to retrieve highly relevant results. (See details in Section 4.)

2. RELATED WORK

In designing an online library system, developers have considered design issues that include (i) formulating and representing users' queries, (ii) processing queries, and (iii) presenting query results. In this section, we discuss existing design problems that library systems are dealing with in terms of (i) performing online searches to retrieve relevant information from library catalogs, (ii) dealing with failed library searches, (iii) ranking retrieved results, and (iv) evaluating the performance of a library system.

As demonstrated by various studies [Lau 2006, Yu 2004], between 10% to 40% of online library catalog searches yield no results. While Yu et al. [Yu 2004] attribute these failed searches to misspellings and typographical errors, wrong selected fields at the time to perform a search (i.e., author, title, etc.), or use of uncontrolled vocabularies (as opposed to LCSH), Lau et al. [Lau 2006] claim that this high percentage of failed searches is due to typographical errors as well as patrons' lack of knowledge in formulating queries that include the proper subject headings associated with a particular library record or Boolean searches. Furthermore, not all of the existing online library systems offer a cross-reference to alternative information, such as thesaurus terms, links to other documents, citation information, etc., to aid users in finding the proper keywords to perform a library catalog search. Thus, users are required to either know the controlled vocabularies provided by the Library of Congress or deal with failures at the time to perform a search.

Another common challenge shared by most of the existing online library systems is the exact-keyword matching which requires keywords as specified (in specific fields such as

title, author, etc.) in the machine readable card (MARC[4]) format associated with a particular library record to be matched exactly with the keywords in a user query in order to locate relevant library catalog records [Larson 1991]. Unfortunately, the exact-match constraint significantly affects the quality and quantity of retrieved results, i.e., irrelevant results or no results at all. *EnLibS*, on the other hand, retrieves relevant library records even though they are not represented by the exact keywords used in a patron's query.

As opposed to most of the library systems that return chronologically ordered results to users, the Endeca-powered catalog at the North Carolina State University Library [Antelman 2006] and the library systems at the state universities in Florida rank retrieved results of a query *Q* by relevance. The ranking strategy (i) places higher the results that share exact keywords in *Q* and in the catalog records, (ii) assigns more weight to results in which the keywords in the query match the keywords in the title rather than other fields, i.e., author, subject, etc., and (iii) uses the term frequency/inverse document frequency (TF-IDF) of keywords in *Q*. RedLightGreen (RLG) [Mattison 2005], on the other hand, proposes a Web-like search alternative with ranking capability, which ranks books based on (i) their relevance to the search terms in a query, i.e., how well the search terms match the terms in the library catalog according to the discussion in [Proffitt 2005], and (ii) the number of libraries that own the books. Unfortunately, as stated in [Mattison 2005], RLG lacks the immediate local holding information which frustrates the targeted audience of RLG. This is because while RLG provides users with a list of relevant results of a search on a library system *S*, there is no guarantee that the library of choice, such as *S,* actually holds the title selected by the user.

A viewpoint in evaluating the performance of a library system is that neither non-empty results nor "zero-hits" results, i.e., no returned results, are solid indicators of a

[4] MARC (http://www.loc.gov/marc/faq.html#definition) is a data format defined by the Library of Congress which allows exchanging, using, and interpreting bibliographic information among computers.

successful or failed catalog search. According to [Cooper 2001], a catalog search can be considered a success not only by conducting controlled experiments or by considering whether library patrons save, print, email, or download a retrieved record, but also by analyzing the *behavior* of a patron during a session. By performing this analysis, such as the time a user spends examining a particular library catalog record, the length of a session, the number of searches performed in a session, the number of title/subject searches in a session, etc., Cooper et al. [Cooper 2001] determine the percentage of sessions, based on the 905,970 sessions conducted on the University of California's Melvyl online library catalog for experimentation, which are considered successful in using a particular library system. Farajpahlou [Farajpahlou 1999] also claims that the success of a library system should not be determined by the success in the performed searches only. Instead, other features, such as the *simplicity* of the system, the *response rate*, and the ability to *coexist* with other library processes, should also be considered. In the study, Farajpahlou [Farajpahlou 1999] proposes to use a 26-item scale criteria for measuring the success of a library system. Although this measure appears to be valid and reliable, a larger number of library systems must be evaluated using the proposed measure before its applicability can be confirmed.

Unlike the searching and ranking approaches mentioned above, we intend to demonstrate that by (i) allowing (in)exact word matches, (ii) detecting semantically similar keywords, and (iii) using representative keywords in a folksonomy (e.g., LibraryThing tags), as opposed to LCSH, to describe a book, we can significantly reduce, if not eliminate entirely, the relatively high percentage of searches that generate no result or irrelevant results and improve the quality and quantity of the results retrieved for a library query.

3. OUR QUERY EVALUATION STRATEGY

In this section, we detail the design of our pre-processing and evaluation strategy for answering library patrons' queries.

3.1. Word Similarity

During the process of evaluating a library patron's query *Q*, we determine the *degree of resemblance* of *Q* and the representation of a library catalog record *R*, which is calculated by using the pre-computed *degrees of similarity* among the keywords in *Q* and *R*. These degrees of similarity, which are the *word-correlation factors* in the word-correlation matrix *M* [Koberstein 2006], were generated by using a set of approximately 880,000 Wikipedia documents (downloaded from http://www.wikipedia.org/), and each factor indicates the *degree of similarity* of two words[5] based on their (i) *frequency* of *co-occurrence* and (ii) relative *distances* in each Wikipedia document. Wikipedia documents were chosen to construct *M*, since they were written by more than 89,000 authors with different writing styles and terminology that cover a wide range of topics. Thus, the Wikipedia documents are rich and diverse in word usage and content. Furthermore, the words in *M* are common words in the English language that appear in various online English dictionaries, such as 12dicts-4.0 (http://prdownloads.sourceforge.net/wordlist/12dicts-4.0.zip), Ispell (http://cs.ucla.edu/geoff/ispell.html), and BigDict (http://packetstormsecurity.nl/Crackers/bigdict.gz).

.1 Word-Correlation Factors

The word-correlation matrix is a 57,908 x 57,908 symmetric matrix, since its word-correlation factors *C(i, j)* and *C(j, i)* are equal, where *i* and *j* are any two given words, and *C (i, j)* reflects how closely related *i* and *j* are, and is defined as

(1)

where $d(w_i, w_j)$ denotes the distance (i.e., the number of words in) between w_i and w_j plus one, *V*(*i*) (*V*(*j*), respectively) denotes the set of words that includes *i* (*j*, respectively) and its stem variations, and |*V(i)*| x |*V(j)*| is the normalization factor. Compared with synonyms

[5] Words in the Wikipedia documents were *stemmed* (i.e., reduced to their grammatical roots) after all the *stop words,* i.e., words with little meaning such as articles, conjunctions, prepositions, etc., were removed which minimize the number of (key)words to be considered. From now on, unless stated otherwise, (key)words refer to *non-stop, stemmed* words.

and related words compiled by *WordNet* (http://wordnet.princeton.edu/), in which pairs of words are not assigned similarity weights, word-correlation factors provide a more *accurate* measure of *word similarity*, which are computed by the appearance of any two words in a huge set of documents. The word-correlation factors in the word-correlation matrix [Koberstein 2006] have been effectively used as a similarity measure in solving various IR problems (see, for examples, [Gustafson 2008] and [Pera 2008]).

Example 1. Using the Harold B. Lee Library (HBLL) system (http://catalog.lib.byu.edu/) at Brigham Young University (BYU) to create the query *Q*: "Climb Alaska" and perform a search against its library catalog, we retrieved no results. Figure 1 shows the HBLL catalog record *R* for the book "Mt. McKinley: the Pioneer Climbs", which is one of the library records that should have been retrieved with respect to *Q,* since *R* describes climbing experiences in Mt. McKinley in Alaska. However, due to the exact-matching evaluation criteria, the book is not retrieved by the HBLL system, which is a major design fault of the library system, as well as various other existing Boolean library systems that process patrons' queries based on exact (-keyword) matching. Table 1 shows the word-correlation factors between the keywords in *Q* and some of the keywords that appear in the *title* and *subject terms* of *R*. Clearly, the non-zero word-correlation factors indicate that keywords in *Q* are related to most of the keywords in *R*, and thus considering the correlation factors of the words in *Q* and *R*, as opposed to exact matches only, it is anticipated that more relevant library catalog records are retrieved with respect to *Q*. □

Due to the size of the word-correlation matrix *M,* which sums up to 6.0 GB, accessing such a huge matrix for determining the possible subset of relevant library records could significantly increase the processing time of *Q*. We consider a reduced version of *M*, which contains 13% of the word-correlation factors of *M* that are the most frequently-occurring words (based on their *frequencies* of occurrence in the Wikipedia documents), and for the remaining 87% of the words only exact-matched correlation factors (i.e., 1.0) are considered. The reduced word-correlation matrix is further minimized to yield the

5×10^{-7}-13% matrix, which retains in the 13% matrix those pairs of words that have a correlation value higher than 5×10^{-7}. Using the further reduced matrix for query evaluation does not affect the accuracy of computing the degree of resemblance of *Q* and *R*, since it contains the top 7,300 most frequently-occurred words that appear in 90% of the Wikipedia documents, and our claim on the accuracy has been verified experimentally. (See Section 4.4 for details.)

.2 Database Records

In order to facilitate the storage structure and query processing techniques offered by existing relational database management systems (RDBMSs), such as query optimization, query execution, scalability, and indexing, we convert the 5×10^{-7}-13% word-correlation matrix into a table in MySQL, called *correlation*5*en*7, which is a three column, 25 MB table that consists of 688,994 tuples. Each tuple is of the form <w_1, w_2, *corrValue*>, where w_1 and w_2 are words, and *corrValue* is the *correlation factor* of w_1 and w_2. In *correlation*5*en*7, w_1 and w_2 form the primary key, and w_1 and w_2 are ordered alphabetically.

.2 Using LibraryThing Tags as Library Record Representation

Instead of considering the keywords of the LCSH specified in a library catalog record *R* in evaluating a library patron's query *Q*, we use an existing *folksonomy* that describes the content of a given object, such as a Web page, a picture, a book, etc. We have chosen the folksonomy defined in *LibraryThing* (http://www.librarything.com) for the illustration purpose in the remaining of this article, since to the best of our knowledge LibraryThing is the most popular social application that was set up solely for cataloging books[6]. However,

[6] The use of LibraryThing tags in academic libraries is discussed in [Ismail 2008], and the benefits of using collaborative tagging to enhance the retrieval quality of an IR system are given in [Clements 2008, Crecelius 2008].

there are other folksonomies available that could also be adopted in implementing *EnLibS*[7]*.* As of February 10, 2009, LibraryThing archives 4,339,326 unique records (on books), and approximately 617,316 users have added more than 46.9 million tags to different book records at LibraryThing, according to the Zeitgeist Overview (http://www.librarything. com/ zeitgeist), which provides official statistical data of LibraryThing.

.2.1 LibraryThing Tags – Valuable Resources for Book Identification

LibraryThing was founded in 2006 for aiding users in cataloging and referencing books. The users of LibraryThing can create an account for rating and reviewing books, as well as adding labels, i.e., *tags*, which describe the content of books in his/her online personal library catalog. A library patron can locate information of books using *commonly-used* and *intuitive* words among LibraryThing tags, rather than the rigidly controlled vocabulary in LCSH specified in library catalog records. Besides serving as a robust cataloging tool, LibraryThing provides a mean of communication among users to share the information archived at personal library catalogs and/or discuss the content of different books, in addition to making book recommendations to others. In other words, LibraryThing uses collective intelligence strategies to suggest books that may (not) be of interest to the users [Starr 2007].

Recall that Figure 1 shows the HBLL catalog record for the book "Mt. McKinley: the Pioneer Climbs", whereas Figure 2(a) shows the corresponding LibraryThing record on the book. While both records share common information, which include title, author, subjects (terms), etc., the LibraryThing record incorporates additional information (as shown in Figure 2(b)), such as reviews, rating, book recommendations, and most importantly a set of *tags* and their respective frequency counts. (Each count of a tag is the total number of users who suggested the tag after reviewing the corresponding book.) Since there is no restriction

[7] In the remaining of this article we consider the folksonomy at LibraryThing; however, we must emphasize that the use of alternative folksonomies would not affect the overall design and performance of *EnLibS*.

on the number of tags that can be used to describe a particular book in LibraryThing, the number of assigned tags of a given book ranges from 1 to thousands. We use the LibraryThing tags to identify each book in a library catalog. Moreover, since LibraryThing includes (i) tags that are personalized and used as a reminder, such as "read," "want to read," "borrowed," as well as (ii) tags that are stop words, which provide little meaning in identifying a library catalog record, these tags are not considered during library query processing[8].

LibraryThing tags were provided by the Harold B. Lee Library as an XML file and could only be used free of charge for research purpose with the proper consent from LibraryThing. Using a script written in the Java programming language to detect each ISBN number that identifies a LibraryThing record *R* in the XML file, each of the tags of *R* is extracted.

.2.2 Reducing the number of LibraryThing Tags and Size of Corresponding Tables

Due to the huge number of tags available on LibraryThing, i.e., 46,920,191 among 4,339,326 library records as of February 10, 2009, we reduce the number of tags to be considered during the query evaluation process by choosing only the top-*n* ($n \geq 1$) tags describing a particular book *B*, where *n* is the top n^{th} *frequency counts* of tags for *B* in LibraryThing. The ideal number of *n* is identified according to (i) its high *accuracy* in retrieving relevant library records as well as (ii) the minimal *processing time* in establishing the degree of resemblance between a query and a library catalog record. In determining the proper value for *n*, we have conducted an empirical study using 100 queries from the HBLL query log on a range of different possible values for *n* and chose the one that satisfies the two criteria listed above. As shown by the conducted experiments, the appropriate value for *n* is *three*.

[8] We have relied on the recommendations made by Tim Spalding, the creator of LibraryThing [MRethlefsen 2007] in identifying personalized and reminder tags. In other words, we remove tags that (i) refer to *where* a particular book might be physically located, (ii) *whom* it belongs to, and (iii) tags that are *not real* words, e.g., "historish".

We include in our MySQL database the *tags* that describe the content of each library catalog record by creating a table *idtag*(*id, tag*), where *id* is a unique identifier of a particular library record *R*, and *tag* is one of the top *three* non-stop, stemmed keywords that represent the content of *R*. *Idtag* is ordered by *id* number, which facilitates the process of locating the descriptive data, i.e., title and author, of a library catalog record[9], which can be retrieved from another MySQL table *catalog*(*id*, *title*, *author*), where *id* is as defined in the *idtag* table. The *idtag* table is 227 MB in size and contains 5,179,553 tuples.

.3 Subset of Relevant Records

The *catalog* table, which contains the HBLL catalog records that match the records found in LibraryThing, can be huge. (As of February 2009, the HBLL catalog includes approximately 3,700,000 records.) It is impractical to evaluate each catalog record against a library patron's query *Q* sequentially. Thus, prior to computing the degrees of resemblance between *Q* and the catalog records, each of which is represented by the *three* most frequent, user-recommended tags, we choose a *subset* of catalog records that are highly likely relevant to *Q.* Each record in the subset must have a tag that is either the same as one of the query keywords in *Q* or their word-correlation factor is at least 3×10^{-5}. In order to facilitate the search of those records that have tags that are the same or highly similar to at least one of the keywords in *Q*, we create another MySQL table, i.e., *tagid*(tag, id), which is ordered alphabetically by *tags* (as opposed to the *idtag* table which contains the same information but is ordered by *id*).

Using 3×10^{-5} as the cut-off value of the word pairs in the 3×10^{-5}-13% matrix, one of the reduced 13% matrices, we select a subset of catalog records to reduce the query processing time without affecting the accuracy of retrieving relevant library records. Figure 3(a) shows the distribution of the word-correlation values among different word pairs in the

[9] Title and Author are the fundamental data provided to a library patron when the latter performs a search.

13% matrix, whereas Figure 3(b) shows the number of word pairs included in various 13% matrices that can be used in the pre-processing step for selecting a subset of library records. Figure 3(b) also shows the *average* query processing time for selecting the corresponding subset of library records for each of the queries in the *HBLL-set,* which consists of a 100 library patron's queries (as partially shown in Table 2) randomly selected from the 2007 HBLL query log. Clearly, it is unacceptable to use the 13% matrix, since it requires an average of 217 seconds in pre-processing each of the queries in *HBLL-set*. Even though the 3×10^{-4}-13% matrix decreases the query processing time to an average of 3.72 seconds, the size of the matrix is reduced by only 620 word pairs compared with the 3×10^{-5}-13% matrix, which on the average requires 4.18 seconds at the pre-processing step. Since the average pre-processing time between the two matrices is *insignificant* and using a matrix with a larger number of word pairs can only enhance the accuracy of retrieving relevant results, we use the 3×10^{-5}-13% matrix, which contains 58,532 word pairs. (The 3×10^{-5}-13% matrix is stored as the table *correlation3en5* in a MySQL database.)

On the average, each word in the 3×10^{-5}-13% matrix is paired with another 1.01 words. Since the average number of keywords included in a user query is 2.35 [Hoscher 2000], it implies that an average of only *three* query keywords are evaluated during the pre-processing step, and the involved processing time ranges between 2 and 5 seconds. Initial experimental results using queries in *HBLL-set* of different sizes (2-4 words) show that the top-10 results, which are often what the users view [Hoscher 2000], are the same when using the 3×10^{-5}-13% matrix compared with using the other matrices as shown in Figure 3(b), which further verifies the effectiveness of using the 3×10^{-5}-13% matrix in terms of *accuracy* and *optimal processing time*.

Example 2. Consider the query *Q*: "Climb Alaska" again. We select the library catalog records against *Q* that have at least one tag that is similar to the query keywords in the *SimWord* column (in the 3×10^{-5}-13% matrix) as shown in Table 3 such that their word-

correlation factors are at least 3×10^{-5}. This selection step reduces the number of possible library records to be considered from 381 to 37. Note that by using the 3×10^{-5}-13% matrix, as opposed to the 3×10^{-6}-13% matrix (as shown in Table 4), we (i) reduce to one-third the total number of similar words to be considered, and (ii) significantly reduce the time required to identify the subset of library records from 19 to 4 seconds without affecting the retrieval of the top relevant library catalog records with respect to *Q*. □

.4 Relevance Ranking

Having selected the subset of library records with respect to a library patron's query *Q*, for each library record R in the subset, we compute the *degree of resemblance* between *Q* and *R*, which is calculated by adding the correlation factors (in the 5×10^{-7}-13% matrix) between each of the *keywords* in *Q* and *tags* (i.e., keywords) associated with *R*. The 5×10^{-7}-13% matrix is used, as opposed to the 3×10^{-5}-13% matrix considered in the pre-processing step, since the former contains the word-correlation factors ($\geq 5\times10^{-7}$) in the 13% matrix, as well as the exact matches for the remaining 87% (as discussed in Section 3.1.1), which provides *more accurate* similarity measure between *Q* and *R* than the more selective 3×10^{-5}-13% matrix. The *degree of resemblance* between *Q* and *R* is defined as

(2)

where *n* (*m*, respectively) denotes the number of keywords in *R* (*Q*, respectively), q_i (r_j, respectively) is a keyword in *Q* (*R*, respectively), and $C(q_i, r_j)$, as defined in Equation 1, is the correlation factor between q_i and r_j in the 5×10^{-7}-13% matrix.

The exact matches (with word-correlation value of 1.0) carry a much heavier weight than other inexact-matched word pairs which are assigned word-correlation factors as low as 5×10^{-7} in the 5×10^{-7}-13% matrix, and thus the *Sim* value of *Q* and *R* is equal to *N* plus a small value, where *N* denotes the number of *exact* matches between *Q* and *R*. As a side effect, the *Sim* function assigns *higher* degree of resemblance to records including tag(s)

that match(es) exactly one or more keywords in Q. As a consequence, if R includes a tag that matches exactly with one of the keywords in Q and has *low* similarity with most of the remaining keywords in Q, then R is ranked higher than a record including tags that are similar (but not exact match) to most of the keywords in Q, which could yield a bias in terms of ranking.

Realizing the shortcomings of *Sim*, we propose another resemblance measure so that if R includes tags *highly similar* to *most* (if not all) of the keywords in Q, then R should be ranked *higher* than another record in which only *one* of its tags is highly similar with only a few of the keywords (or matches exactly one keyword) in Q. An alternative measure of the *degree of resemblance* between Q and R is defined as

(3)

where Q, R, n, m, C, q_i, and r_j are as defined in Equation 2.

By using the *Min* function in Equation 3, we impose a constraint on summing up the correlation factors of keywords in Q and R. Even if a tag in R (i) matches exactly one of the keywords in Q and (ii) is similar to some of the remaining keywords in Q (which would yield a value greater than 1.0, the word-correlation factor of an exact match), we limit the sum of their word-correlation factors to 1.0. This constraint ensures that if R contains a *dominant* tag T, i.e., T is similar to (or the same as) a few keywords in Q, T alone cannot significantly impact the resemblance value of R and Q, whereas if R contains a number of tags that are similar to most of the keywords in Q, then R is assigned a *higher* degree of resemblance due to its *diversity* in matching keywords in Q.

Example 3. Consider the query Q defined in Example 1. Table 5 (Table 6, respectively) shows 10 (out of the 37) retrieved catalog records and their degrees of resemblances with Q computed by using the *Sim* (*LimitedSim*, respectively) measure. Table 7 shows the titles of the records in Tables 5 and 6. By restricting the sum of the word correlation factors between a tag in R and all the keywords in Q to 1.0 using Equation 3, a

comparatively higher degree of resemblance (i.e., *LimitedSim* value) is assigned to library catalog records which include tags that match most of the keywords in *Q*. Even though Record 3 in Table 6 has a *lower* similarity value (computed by using Equation 3) with respect to the same record in Table 5 (computed by using Equation 2), Record 3 is ranked higher, i.e., at the fourth position, in Table 6, since its keywords are similar to both keywords in *Q* (i.e., climb and Alaska), which indeed is *more* relevant in terms of its content than the fourth ranked record, i.e., Record 6, in Table 5, which is similar to *only* one of the keywords in *Q* (i.e., Alaska), and the contents of Records 3 and 6 have been verified manually. Moreover, records that are related (in term of their contents with respect) to *Q*, such as Records 1, 2, and 4, are ranked *higher* (i.e., at positions 9, 7, and 8, respectively) by *LimitedSim*, whereas the same records are ranked *lower* (17, 16, and 15, respectively) by *Sim*.

.5 Query Processing Time

As stated earlier, one of the design goals of *EnLibS* is to process user queries with processing time compatible with existing library search engines. In an attempt to reduce the query processing time, we have constructed sophisticated data/file structures for storing (i) general information about library catalog records, (ii) LibraryThing's tags describing the content of library catalog records, and (iii) the reduced word-correlation matrices (i.e., the *correlation*3*en*5 and *correlation*5*en*7 MySQL tables), besides using the *InnoDB* storage engine of the MySQL database, which is designed for maximizing the performance in processing large data volumes and has a CPU efficiency that is not matched by other disk-based relational database engines (see http://dev.mysql.com/doc/refman/5.1/en/innodb-overview.html).

.5.1 Prefix-string Indexes

Since significant query processing time is allocated for selecting the proper subset of library catalog records with respect to a library patron's query (as discussed in Section 3.3),

we have implemented alternative prefix-string indexes (besides the primary indexes) on *correlation*3*en*5, *correlation*5*en*7, and *tagid* tables. In creating the prefix-string indexes on these tables, we follow the recommendations made by [Dubois 2005] who claim that (i) since *shorter* values are compared more *quickly*, implementing prefix-string indexes on *smaller* index values as opposed to indexing the entire column allows *faster* lookups, (ii) *smaller* indexes require *less* disk access, and (iii) by considering *shorter* indexing values, MySQL can hold *more* keys in the cache memory, which translates into *less* index blocks swapping from disks in performing a search, a major bottleneck in query processing. Hence, we use a pre-determined prefix length in defining a prefix-string index for the corresponding columns in *correlation*3*en*5, *correlation*5*en*7, and *tagid* tables, instead of indexing the entire columns in the tables.

In creating an index on a string column, Dubois [Dubois 2005] suggests indexing 10% of the entire length of the column. Based on these recommendations and since the tag (word, respectively) in the *tagid* (*correlation*3*en*5 and *correlation*5*en*7, respectively) table is between 20 and 25 characters long, we define a prefix-string index on the string prefix of length 3 in the "word" column in the *correlation*3*en*5 and *correlation*5*en*7 tables, and the "tag" column in the *tagid* table.

.5.2 Query Processing Time/Memory Allocation for Indexing

We have verified the appropriateness of choosing the three-character prefix strings as the prefix-string index values. Figure 4(a) shows (i) the average time (in seconds) for processing the queries in *HBLL-set* using prefix-string indexes of different prefix sizes, i.e., 3, 5, and 8 characters, as well as (ii) the memory space required for these prefix-string indexes. Although the difference between the average query processing time when using the prefix-string index of size 3 instead of size 5 is not significant (7.0 versus 7.8 seconds), the required memory space is reduced significantly (from 195.4 MB to 181.3 MB), which further confirms the ideal choice of using the three-character prefix-string indexes. Furthermore,

the subset of catalog records chosen at the pre-processing step does not change when prefix-string indexes of different sizes are implemented. Hence, the *accuracy* of the retrieval is *not* compromised when using *shorter* (instead of longer) prefix-string indexes. Table 8 shows the size (in MB) of each indexed table in our MySQL database, as well as the size (in MB) of the corresponding prefix-string indexes, whereas Figure 4(b) shows the average processing time required to answer a query, with and without using the (top-3) tags and prefix-string indexes on the queries, in *HBLL-set*. Note that *idtag* is not included in Table 8 since it is indexed by *id*, which does not require the use of prefix-string indexes. Due to the significant processing time reduction (from 429 to 7 seconds), the choice of using the (i) prefix-string indices, (ii) the 3×10^{-5}-13% matrix, and (iii) top-three LibraryThing tags is obvious.

Furthermore, in our pre-processing step the subset of selected library catalog records contain tags that match exactly or are highly similar to the keywords in a user query. The number of highly similar keywords in the records (with respect to the keywords in a user's query) determines the number of records to be further ranked. Moreover, the *more* records retrieved, the *higher* the number of records to be evaluated to determine their degrees of resemblance with respect to a user's query, and the *longer* query processing time is required. By using the 3×10^{-5}-13% matrix in processing the queries in *HBLL-set*, it has been shown that the average number of similar query keywords and the original query keywords to be compared with *LibraryThing* tags is 9, as opposed to 200, if the 5×10^{-7}-13% matrix is used instead. More importantly, the reduced number of keywords to be compared does *not* affect the quality of the retrieved results, which has been verified manually.

.6 The Overall Evaluation Process

Figure 5 shows the entire query evaluation process of *EnLibS*, which illustrates that when a library patron submits a query *Q*, keywords in *Q* are first reduced to their

grammatical roots and stop words are eliminated, i.e., step (i). Using the set of non-stop, stemmed keywords *K* and the *correlation*3*en*5 table, we retrieve the set of correlated keywords *SK* in the table, including the keywords in *K*, i.e., step (ii), which are matched with the tags that describe each of the library catalog records in the *tagid* table, and the matching yields the subset *S* of library catalog records that are highly likely relevant to *Q*, i.e., step (iii). Hereafter, using the *idtag* table we identify each record in *S* and based on their tags (i.e., the top-three LibraryThing's tags associated with a particular record, based on their frequency count) along with the *word-correlation factors* from the *correlation*5*en*7 table, we rank the retrieved records in *S* according to their *degrees of resemblance* with *Q*, i.e., step (iv).

4 EXPERIMENTAL RESULTS

In this section, we assess the performance of *EnLibS*. We first describe the dataset used for the experiments and the evaluation strategies adapted for performance analysis. Hereafter, we present the percentage of library patron's searches in the dataset that yield zero-hits, i.e., no results, and compare the performance of our similarity matching and ranking approach in retrieving relevant results with the one achieved by the BYU HBLL system.

4.1 The Dataset

In evaluating the performance of *EnLibS* in querying library catalogs, we used the queries in the HBLL *query log* created between July 2006 and January 2007, a file that is 144 MB and the average size of each entry in the log is 180 bytes. Each entry in the log includes (i) a query, (ii) date and time when the query was formulated, and (iii) the corresponding number of records retrieved. Due to the large number of queries in the log (approximately one million queries), we randomly selected 320 of them, which constitute the test set, denoted *HBLL-log*, for analyzing the performance of *EnLibS*. The queries in the

HBLL query log (and hence *HBLL-log*), which were formulated during the 7-month period and on an average contain 2.45 non-stop, stemmed words, cover a wide variety of subject areas that include Biology, Computer Science, Education, Geography, Mathematics, Medicine, Music, Religion, etc. Due to the (i) large quantity of queries in the HBLL query log, (ii) the diversity of users who formulated the queries, and (iii) the general subject areas covered in the queries, *HBLL-log* is an ideal dataset for the empirical study[10].

4.2 The Evaluation Methods

In order to analyze the accuracy of the retrieved and ranked library records of each library patron's query in *HBLL-log*, we rely on measures commonly used for determining the effectiveness of information/data retrieval systems, i.e., precision[11]. *Precision* determines the fraction of *retrieved* records that are *relevant*, which quantifies the set of library records retrieved by *EnLibS* using LibraryThing tags, as opposed to the HBLL system using LCSH, in processing a query. In general, library patrons view only the first 10 retrieved results when performing a search [Hoscher 2000], and hence we considered the first 10 retrieved records (if they exist) for each query in *HBLL-log*. We have adapted the precision measure in [Goncalves 2004] to compute the 10-*Precision* value, which quantifies the top-10 retrieved results in terms of their relevance with respect to a query *Q*, and is defined as

(4)

where *#_of_Retrieved_Relevant_Records* denotes the number of relevant records with respect to *Q* in the top-10 retrieved results.

Furthermore, in providing additional performance evaluation of *EnLibS*, we consider

[10] To the best of our knowledge, there are no benchmark datasets/measurements that can be used for evaluating the retrieval and ranking performance of any library system.

[11] The dataset used in our empirical study, i.e., library records in the HBLL *query log* (and thus the *HBLL-log*), have not been previously labeled as (ir)relevant with respect to each query in its set, and hence the *recall ratio* cannot be determined in this study, which is not as significant as *precision* in measuring the top-ranked retrieved records, which are the ones library patrons examine.

the *Mean Average Precision* (*MAP*) [Aslam 2006], which is defined as

(5)

where Q is the total number of queries in a dataset, i.e., *HBLL-log* in our case, r is the number of relevant documents to be considered, and $t(q)$ is the total number of records retrieved when the r^{th} relevant record on the q^{th} query is encountered. By using *MAP*, we can also determine the effectiveness of our ranking approach in terms of positioning higher in the rank the documents with higher degrees of relevance [Baeza-Yates 1999], in addition to compute the 10-precision value of *EnLibS*

We have manually examined each of the retrieved records, up till the requested number of relevant ones to be retrieved, for each query Q in *HBLL-log* and labeled them as either relevant or irrelevant with respect to Q, which generated the 10-*Precision* and *MAP* performance measures.

4.3 Queries with Zero-hits

In Section 1, we have discussed one of the shortcomings of existing library systems---the large percentage of *zero-hits*, i.e., library patron's queries that yield no results. With that in mind, we have designed *EnLibS* to minimize the number of zero-hits using *word-similarity matching* and *folksonomies* from LibraryThing. To verify that this design goal is achieved, we compared the number of queries in *HBLL-log* that yield zero-hits using the HBLL system, as well as *EnLibS*. According to the experimental results, the percentage of zero-hits searches is reduced from 16% (using the HBLL system) to 1.6% (using *EnLibS*), i.e., from 51 to 5 zero-hits, which is a significant improvement (see Figure 6). Most importantly, the (top 10) results retrieved by *EnLibS* for each of the queries for which the HBLL library system retrieved no results at all were manually examined, and the examination showed that *EnLibS* retrieved *relevant results* for queries that the HBLL system yielded zero-hits.

4.4 The Overall Accuracy

Besides determining the number of zero-hits, we have evaluated the overall accuracy of *EnLibS* using the 10-*Precision* of the top-10 results, which measures the correctness of the (ranking on the) retrieved results. Figure 7(a) shows that for each query in *HBLL-log*, its 10-*Precision* value of the results retrieved by using *EnLibS* is higher than the 10-*Precision* value generated by using the HBLL system. What is more, *EnLibS* achieves an average of 0.74 10-*Precision* as opposed to the average of 0.42 10-*Precision* obtained by the HBLL system, which shows that on the average between seven and eight of the top-10 results retrieved by *EnLibS* are relevant with respect to a patron's query, compared with about four results retrieved by the HBLL system.

Example 4. Table 9 shows the top-10 library records (identified by their titles) retrieved for the query "Apartheid" using the HBLL system and *EnLibS,* respectively. *EnLibS* retrieves nine relevant records (when considering only the top-10 retrieved ones), which are highlighted, as opposed to the four retrieved by the HBLL system. Furthermore, when using *EnLibS* relevant records are positioned higher in the ranking. □

Although we measure the accuracy of *EnLibS* based only on the top-10 results of each test query in HBLL-log, *EnLibS* actually retrieves more than the top-10 results, if they exist, for each query. In fact, *EnLibS* retrieves as many library records, which are treated as relevant in the proper subset (as discussed in Section 3.3) with respect to each library patron's query *Q*, which could be in the hundreds and ordered according to their degrees of resemblance to *Q.* Of course the lower the position of a library record in the ranking, the less relevant the record is to *Q*.

We have also evaluated the performance of *EnLibS* in terms of the *MAP* measure. In comparing the *MAP* values generated by the HBLL system and *EnLibS*, we set r = 3, 5, 7, and 10, i.e., evaluated the top-3, top-5, top-7, and top-10 relevant records retrieved by using the queries in *HBLL-log*, respectively. As shown in Figure 7(b), the *MAP* values obtained by *EnLibS* are higher than the corresponding ones obtained by the HBLL system. A

higher MAP value means that *less* library records are accessed in finding the desired number (i.e., *r*) of relevant records. According to the experimental results, on the average *EnLibS* locates the $r \in \{3, 5, 7, 10\}$ desired relevant records between the $r + 1^{th}$ and $r + 3^{th}$ record, whereas the HBLL system requires an average between the $r + 2^{th}$ and $r + 8^{th}$ record.

4.5 Query Processing Time

We have assessed the performance of our enhanced library system in terms of *processing time* by measuring on the average the amount of time required to evaluate each query in *HBLL-log*. When processing the queries in *HBLL-log*, the average time required for processing each one of the queries using the HBLL system is 6.1 seconds, whereas by using *EnLibS* the average time is 7.0 seconds.

Although the query processing time of *EnLibS* is higher than the query processing time of the HBLL system, the difference, which is less than one second, is not significant, especially when the results retrieved by *EnLibS* are more accurate, in terms of relevancy and ranking, than the results generated by the HBLL system.

4.6 Impacts of *EnLibS*

Searches performed by using the HBLL system are powered by *SirsiDynix*'s *Unicorn*, which is installed on a significant number of library systems at different places around the world, e.g., Arizona State Library, Carnegie Mellon University, Gribskov Community Library-Denmark, Kansas City Public Library, Natural Resources Canada Library, Pennsylvania State University, Princeton City Schools, Supreme Court of Canada Library, to name a few (see Unicorn's official Web site http://www.sirsidynix.com/Solutions/Products/integratedsystems.php). While *Unicorn* includes necessary features such as modules for circulation, acquisitions, outreach, materials booking, reserves, etc., it still lacks the *accuracy* in retrieving and ranking relevant library catalog records. By incorporating (i) the use of LibraryThing tags, (ii) similarity matching, and (iii) relevance ranking, we enrich the catalog

searches powered by *Unicorn* and hence the library systems used by many private and public libraries.

5 CONCLUSIONS

Library catalogs offer library patrons a mean to locate the extensive resources available in public and private libraries. Unfortunately, due to the high percentage of searches that yield irrelevant or no results and the lack of relevance ranking, in addition to the difficulty in formulating queries using the rigid and unintuitive keywords in library catalog records, which are defined by the Library of Congress Subject Heading to perform an exact keyword(-matching) search, library patrons have been turning to Web search engines to locate the initial information (such as titles, authors, subject areas, etc.) first, which yield the primitive information that library patrons can later use for querying the library catalog, a tedious and inefficient searching strategy.

In order to improve existing library searches, we have proposed to use *word-correlation factors* and *folksonomies* to perform *similarity matches* between keywords in a library patron's query and the user-generated tags from *LibraryThing*, which describe the contents of library books in library catalogs using commonly-used words. Experimental results show that the proposed library system, *EnLibS*, (i) significantly reduces zero-hits query results and (ii) ranks highly relevant library records high by using our similarity matching and degree of resemblance approach, while maintaining the processing time comparable with existing library search engines. *EnLibS* outperforms and can be adapted for enhancing existing library systems powered by the search engine of *SirsiDynix's Unicorn*, a widely-used integrated library system at private and public libraries these days.

Regarding future work, we would like to further enhance the performance and types of queries that can be handled by *EnLibS.* We plan to incorporate a Fuzzy Set IR model evaluation strategy on *EnLibS* to handle Boolean queries, i.e., *EnLibS* users can formulate

their queries using Boolean operators, such as AND, NOT and OR. By using these Boolean operators, *EnLibS* users can create more complex queries, if they so desire, which allow the specification of inclusion, exclusion, and alternation of keywords as an advanced search option (which is available among popular Web and library search engines) that can enhance the expressive power of *EnLibS*.

Furthermore, we will consider *scaling* the values of the word-correlation factors used for computing the degree of similarity among the keywords in a query and the tags that describe a particular library record, since by replacing the currently-used, word-correlation factors, which are in the range of $3x10^{-5}$ or lower, for their corresponding scaled values between 0% and 100%, we can provide a more intuitive, i.e., easier to understand, similarity value for determining the probability that any two words share the same semantic meaning. Using the scaled word-correlation factors, the accuracy of *EnLibS* should not be affected.

Moreover, we would also like to continue our study by assessing the use of folksonomies compared to controlled vocabularies, i.e., the correlation between tags and subject headings associated with a particular library catalog record, with the purpose of using the most appropriate LibraryThing tags for representing a library record (even if they are not the most frequently occurring ones).

REFERENCES

[Antelman 2006] Antelman, K., Lynema, E., & Pace, A. (2006). Toward a 21st Century Library Catalog. Information Technology and Libraries, 25(3), 128–139.

[Aslam 2006] Aslam, J., Pavlu, V. & Yilmaz, E. (2006). A Statistical Method for System Evaluation Using Incomplete Judgments. In Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval (SIGIR), pp. 541–548.

[Baeza-Yates 1999] Baeza-Yates, R. & Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison Wesley.

[Borgman 1996] Borgman, C. (1996). Why are Online Catalogs Still Hard to Use? Journal of the American Society for Information Science (JASIS). 47(7), 493-503.

[Clements 2008] Clements, M., de Vries, A., & Reinders, M. (2008). Detecting Synonyms in Social Tagging Systems to Improve Content Retrieval. In Proceedings of ACM Conference on Research and Development in Information Retrieval (SIGIR), pp. 739-740.

[Cooper 2001] Cooper, M. & Chen, H. (2001). Predicting the Relevance of a Library Catalog Search. Journal of the American Society of Information Science and Technology (JASIST), 52(10), 813–827.

[Crecelius 2008] Crecelius, T., Kacimi, M., Michel, T., Neumann, T., Parreira, J., Schenkel, R., & Weikum, G. (2008). Making SENSE: Socially Enhanced Search and Exploration. In Proceedings of Very Large Data Bases (VLDB), pp. 1480-1483.

[De Rosa 2005] De Rosa, C., Cantrell, J., Cellentani, D., Hawk, J., Jenkins, L., & Wilson, A. (2005). Perceptions of Libraries and Information Resources. In Online Computer Library Center (OCLC ’05), Technical Report.

[Dubois 2005] Dubois, P. (2005). MySQL - 3rd Edition. Published by Sams as part of the

Developer's Library Series.

[Farajpahlou 1999] Farajpahlou, A. (1999). Defining Some Criteria for the Success of Automated Library Systems. Library Review, 48(4), 169–180.

[Goncalves 2004] Goncalves, M., Fox, E., Krowne, A., Calado, P., Laender, A., da Silva, A. & Ribeiro-Neto, B. (2004). The Effectiveness of Automatically Structured Queries in Digital Libraries. In Proceedings of the 4th ACM/IEEE-CS joint conference on Digital Libraries (JCDL'04), pp. 98–107.

[Gustafson 2008] Gustafson, N., Pera, M.S., & Ng, Y.-K. (2008). Nowhere to Hide: Finding Plagiarized Documents Based on Sentence Similarity. In Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'08), pp. 690-696.

[Herrera 2007] Herrera, G. (2007). MetaSearching and Beyond: Implementation, Experiences and Advice from an Academic Library. Information Technology and Libraries, 26(2), 44–52.

[Hoscher 2000] Hoscher, C. & Strube, G. (2000). Web Search Behavior of Internet Experts and Newbies. Computer Networks: The International Journal of Computer and Telecommunications Networking, 33, 337–346.

[Inouye 2001] Inouye, A. (2001). A Digital Strategy for the Library of Congress. Communications of the ACM (CACM), 44(5), 43.

[Ismail 2008] Ismail, L. (2008). LibraryThing for Libraries. The Charleston advisor, 10(2), 23-26.

[Koberstein 2006] Koberstein, J. & Ng, Y.-K. (2006). Using Word Clusters to Detect Similar Web Documents. In Proceedings of Knowledge Science, Engineering and Management (KSEM '06), pp. 215–228.

[Larson 1991] Larson, R. (1991). The Decline of Subject Searching: Long-Term Trends and Patterns of Index Use in an Online Catalog. Journal of the American Society of

Information Science and Technology (JASIST), 42(3), 197–215.

[Larson 1992] Larson, R. (1992). Evaluation of Advanced Retrieval Techniques in an Experimental Online Catalog. Journal of the American Society of Information Science and Technology (JASIST), 43(1), 34–53.

[Lau 2006] Lau, E. & Goh, D. (2006). In Search of Query Patterns: A Case Study of a University OPAC. Information Processing and Management, 42(5), 1316–1329.

[Li 2008] Li, H. & Deng, S. (2008). Linking Location and Shelf Mapping from OPAC Search Results: with Reference to Wichita State University. New Library World, 109(3/4), 107–116.

[Mattison 2005] Mattison, D. (2005). RedLightGreen and Open WorldCat. Searcher, 13(4), 14-23.

[MRethlefsen 2007] Rethlefsen, M. (2007). LJ talks to Mastermind of the LibraryThing Web Site, Bookhound Tim Spalding. Library Journal (http://www.libraryjournal.com /article/ CA6403633.html).

[Neal 2007] Neal, D. (2007). Introduction: Folksonomies and Image Tagging: Seeing the Future? Bulletin of the American Society of Information Science and Technology, 34(1), 7–12.

[Novotny 2004] Novotny, E. (2004). I Don’t Think I Click: A Protocol Analysis Study of Use of a Library Online Catalog in the Internet Age. College and Research Libraries, 65(6), 525–537.

[Pera 2009] Pera, M.S. & Ng, Y.-K. (2009). SpamED: A Spam Email Detection Approach Based on Phrase Similarity. Journal of the American Society for Information Science and Technology (JASIST), 60(2), 393-409.

[Proffitt 2005] Proffitt, M. (2005). How to ‘Google’ Your Library Catalog. NLA News, XV(4). National Library of Australia. Available: http://www.nla.gov.au/pub/nlanews/2005 /

jan05/article1.html.

[Rethlefsen 2007] Rethlefsen, M. (2007). Tags Help Make Libraries Del.icio.us: Social Bookmarking and Tagging Boost Participation. Library Journal, 15, 26–28.

[Starr 2007] Starr, J. (2007). LibraryThing.com: The Holy Grail of Book Recommendation Engines. Searcher, 15(7), 25-32.

[Yu 2004] Yu, H. & Young, M. (2004). The Impact of Web Search Engines on Subject Searching in OPAC. Information Technology and Libraries, 23(4), 168–180.

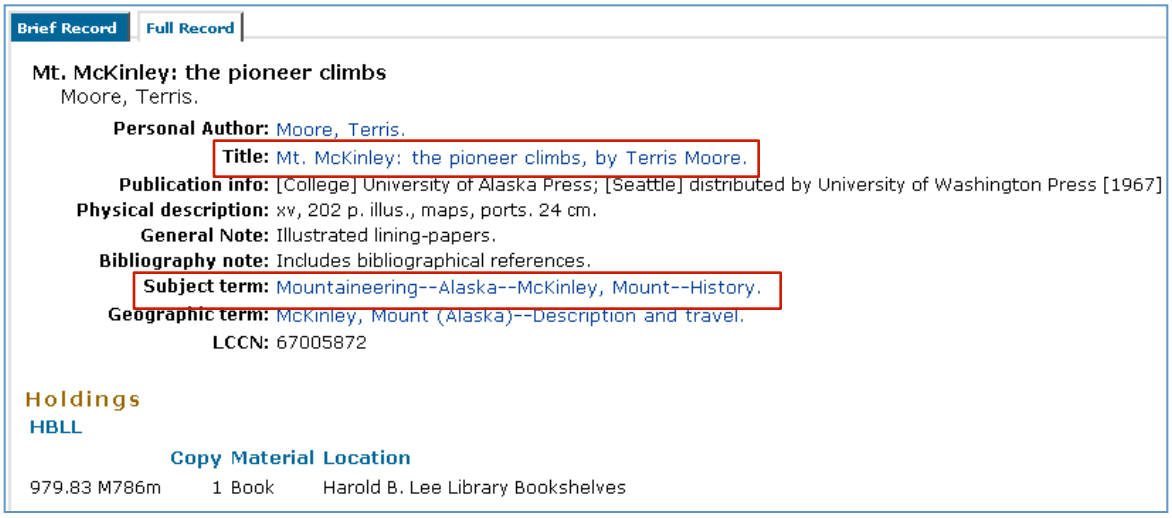
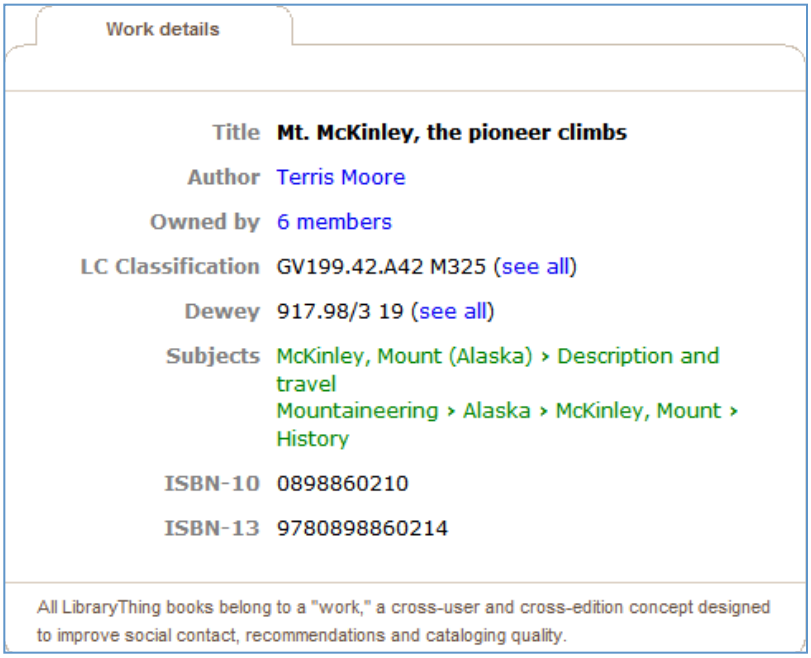
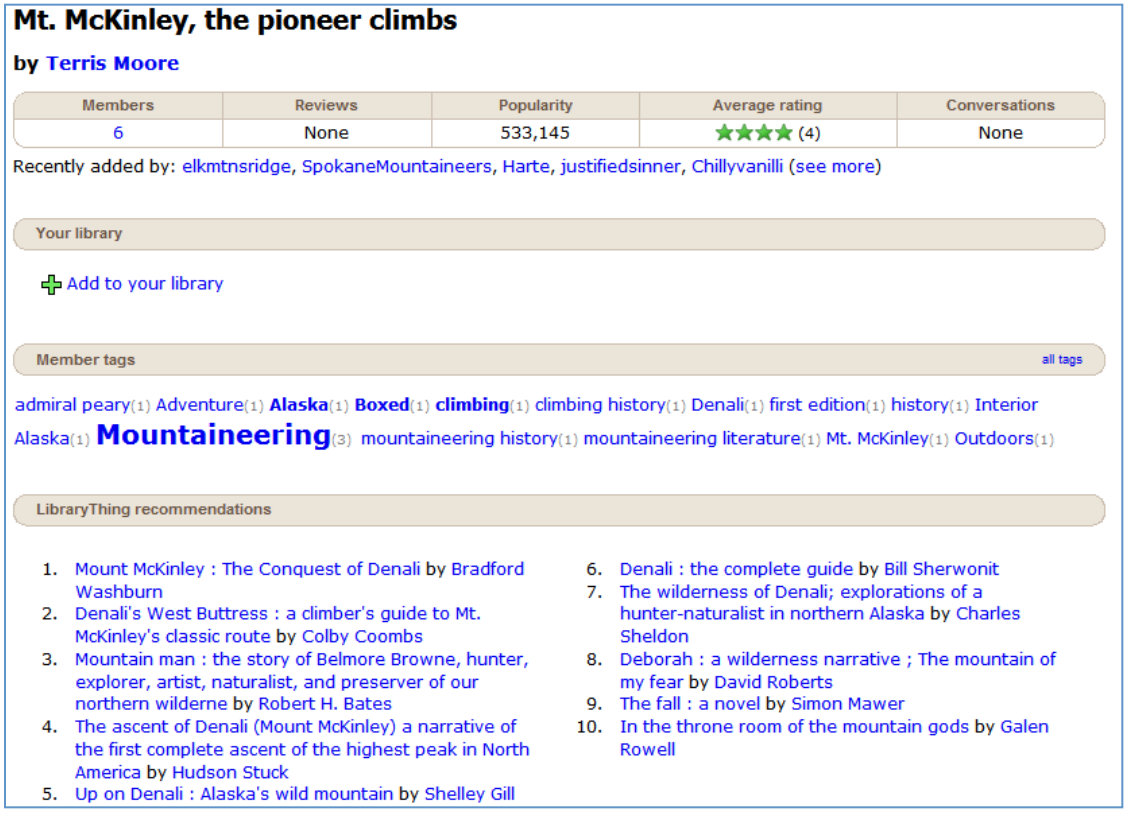


Figure 1: HBLL library catalog record *R for* the book "Mt. McKinley: the Pioneer Climbs"



(a) Information on the book "Mt. McKinley: the Pioneer Climbs" as shown in the corresponding LibraryThing record



Mt. McKinley, the pioneer climbs

by Terris Moore

Members	Reviews	Popularity	Average rating	Conversations
6	None	533,145	★★★★ (4)	None

Recently added by: elkmtnsridge, SpokaneMountaineers, Harte, justifiedsinner, Chillyvanilli (see more)

Your library

Add to your library

Member tags all tags

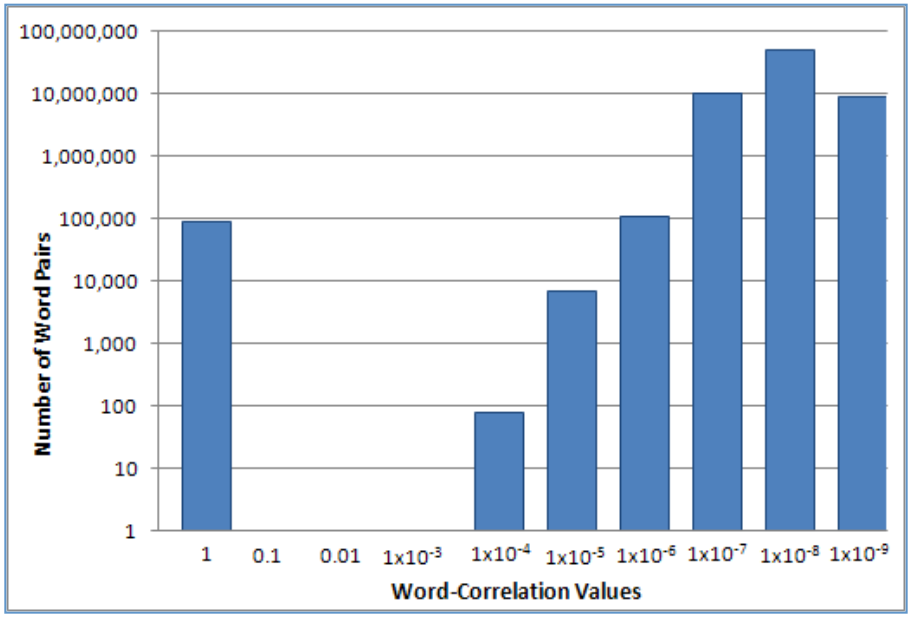
admiral peary(1) Adventure(1) **Alaska**(1) **Boxed**(1) **climbing**(1) climbing history(1) Denali(1) first edition(1) history(1) Interior Alaska(1) **Mountaineering**(3) mountaineering history(1) mountaineering literature(1) Mt. McKinley(1) Outdoors(1)

LibraryThing recommendations

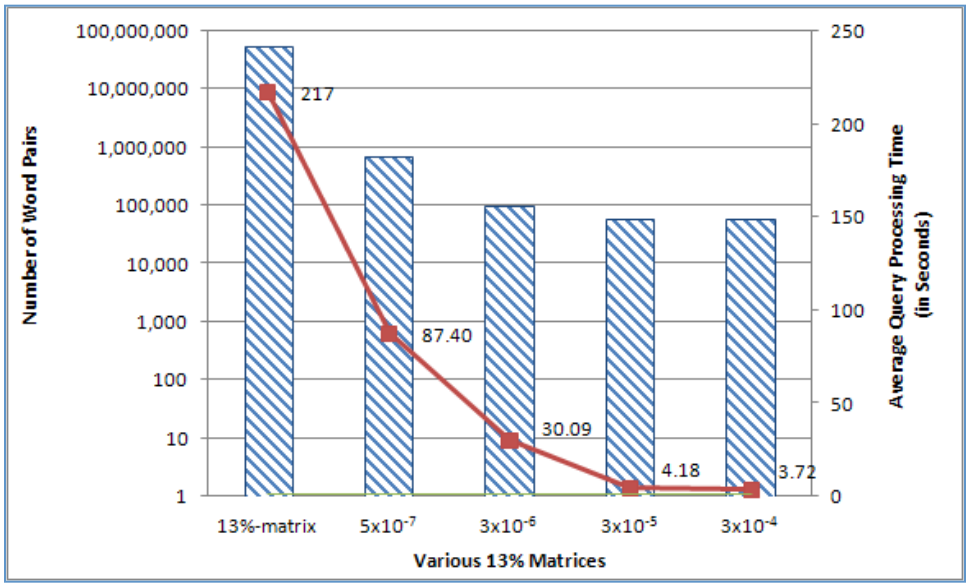
1. Mount McKinley : The Conquest of Denali by Bradford Washburn
2. Denali's West Buttress : a climber's guide to Mt. McKinley's classic route by Colby Coombs
3. Mountain man : the story of Belmore Browne, hunter, explorer, artist, naturalist, and preserver of our northern wilderne by Robert H. Bates
4. The ascent of Denali (Mount McKinley) a narrative of the first complete ascent of the highest peak in North America by Hudson Stuck
5. Up on Denali : Alaska's wild mountain by Shelley Gill
6. Denali : the complete guide by Bill Sherwonit
7. The wilderness of Denali; explorations of a hunter-naturalist in northern Alaska by Charles Sheldon
8. Deborah : a wilderness narrative ; The mountain of my fear by David Roberts
9. The fall : a novel by Simon Mawer
10. In the throne room of the mountain gods by Galen Rowell

(b) Tags and additional information created by different LibraryThing users for the book "Mt. McKinley: the Pioneer Climbs"

Figure 2: Information associated with the book "Mt. McKinley: the Pioneer Climbs" available on LibraryThing

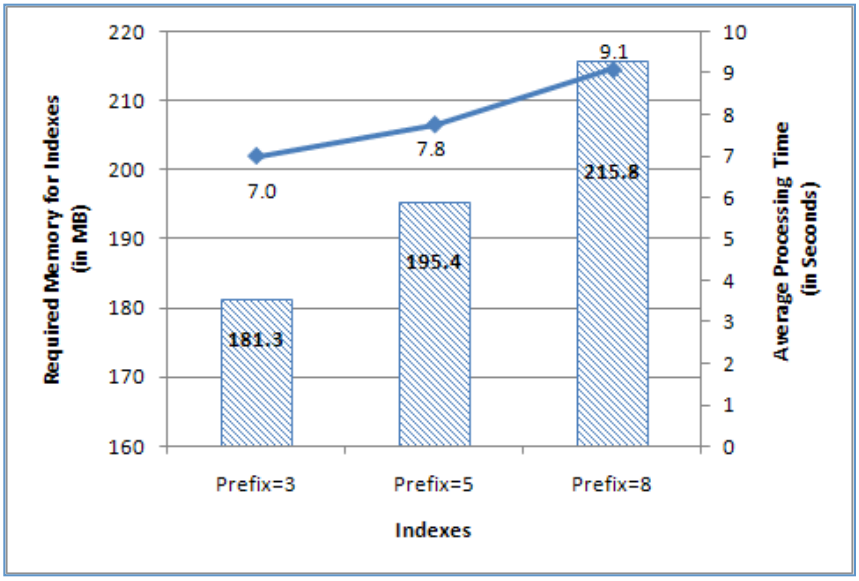


(a) Distribution of the word-correlation values in the 13% matrix

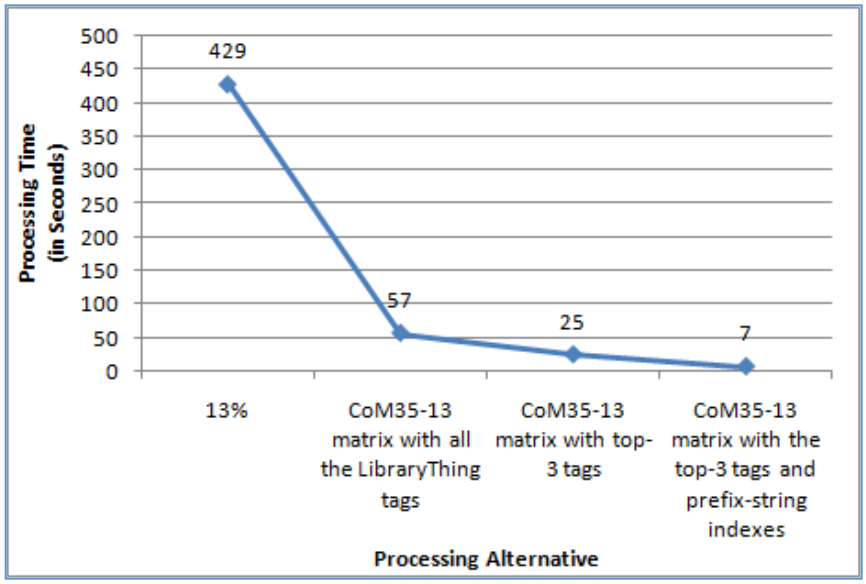


(b) Number of word pairs in each 13% matrix and the average processing time of the *HBLL-set*

Figure 3: Word-correlation factors in the 13% matrix and the query processing time using different variations of the 13% matrix



(a) Required memory allocation for prefix-string indexes and their average processing time on *HBLL-set*



(b) Average processing time in answering queries in *HBLL-set* using different data and indexing tables

Figure 4: Query processing time and memory sizes of indexed tables used in *EnLibS*

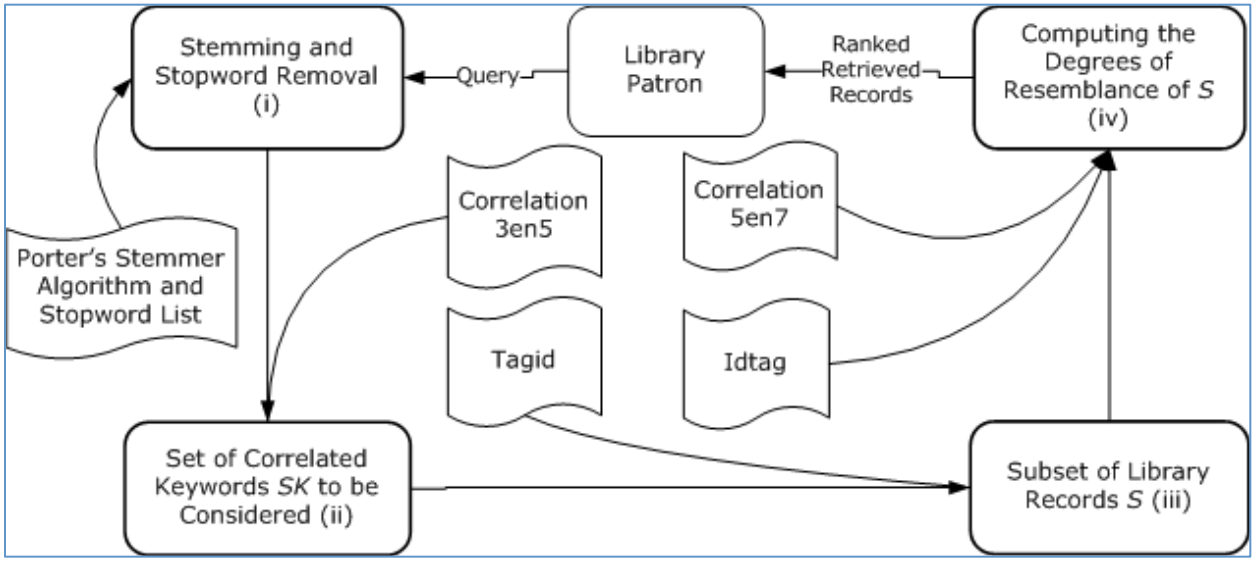


Figure 5: The overall query evaluation process of *EnLibS*

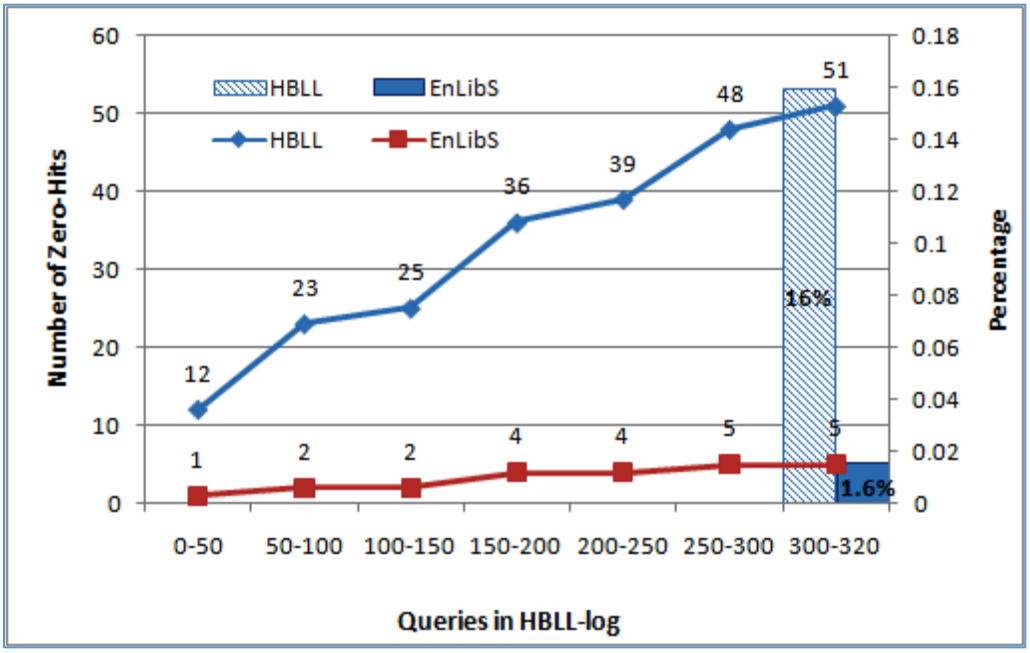
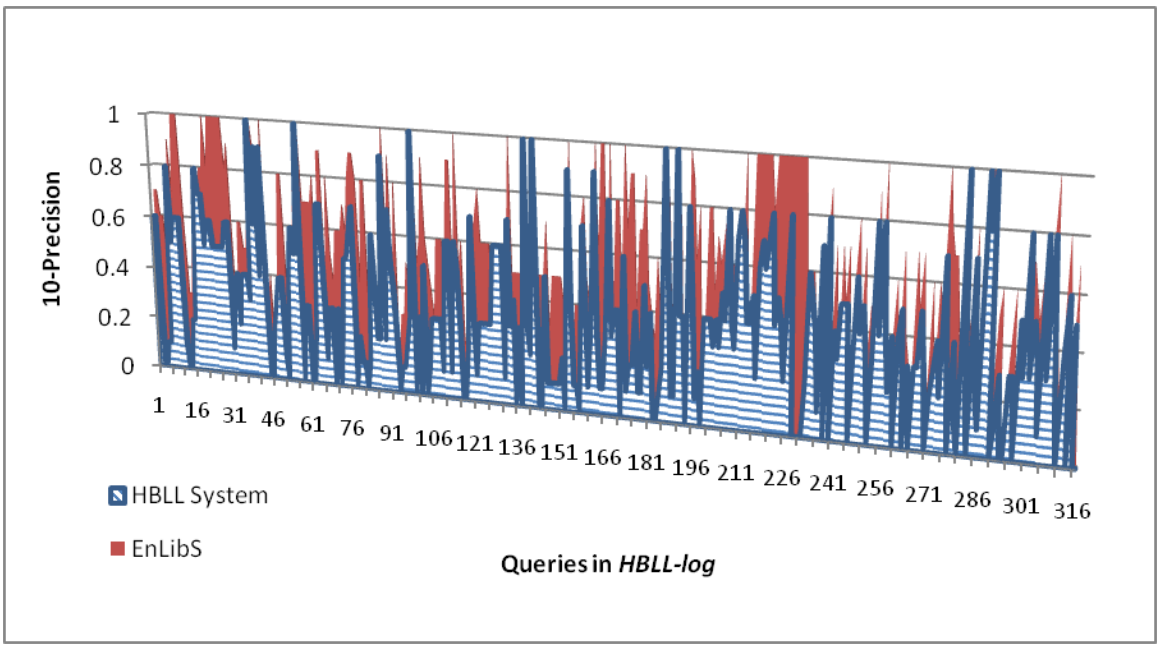
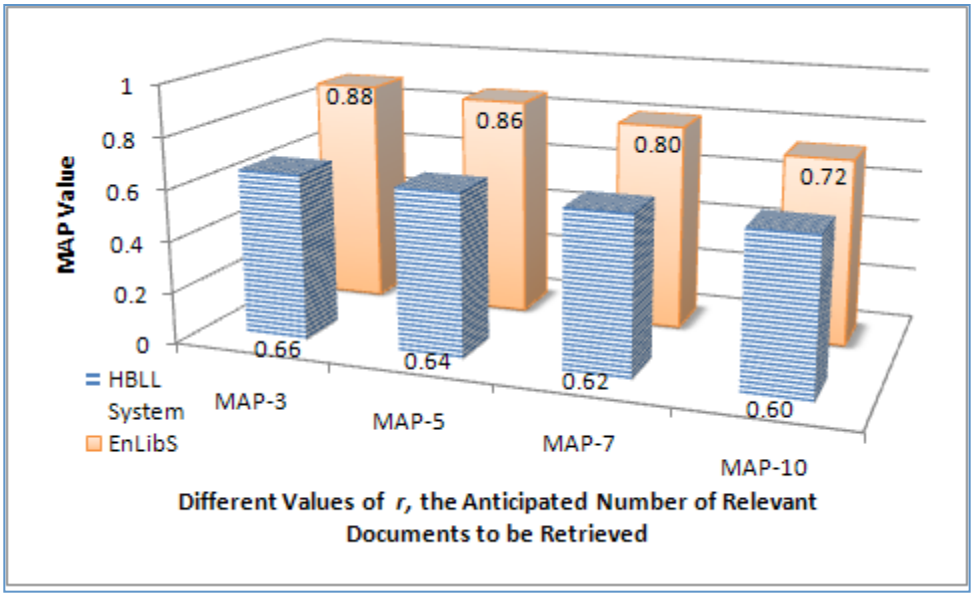


Figure 6: Zero-hits queries in *HBLL-log* generated by the HBLL system and *EnLibS*



(a) The 10-*Precision* measures



(b) *MAP* measures on $r \in \{3, 5, 7, 10\}$ values

Figure 7: Performance evaluation on *EnLibS* using the *HBLL-log* queries

	Alaska	climb	history	McKinley	mount	mountaineering
climb	$7.5x10^{-5}$	1.0	$9.3x10^{-8}$	0.0	$6.2x10^{-5}$	$2.5x10^{-6}$
Alaska	1.0	$7.5x10^{-5}$	$3.7x10^{-8}$	0.0	$4.5x10^{-5}$	$1.4x10^{-5}$

Table 1: The word-correlation factors of (portion of the) keywords in the query *Q*: "Climb Alaska" and the title and subject terms of the library record *R* as shown in Figure 1

	Query	Stemmed, Non-Stop Keywords
1	Mennonite Hymn	mennonit hymn
2	parental death	parent death
3	distance run	distanc run
4	lou gehrig	lou gehrig
5	catch me if you can video recording	catch video record
6	teach yourself guitar	teach yourself guitar
7	soldier's report	soldier report
8	phrenology neuroscience	phrenolog neurosci
9	freezer substitution	freez substitut
10	minimalistic bible	minimalist bibl
11	win child custody war	win child custodi war

Table 2: A subset of library patron's queries in *HBLL-set* and their corresponding stemmed, non-stop word versions used for experimentation

Word	Similar Word	Correlation Factor
alaska	alaska	1.0
alaska	climb	$7.5x10^{-5}$
alasta	mount	$4.5x10^{-5}$
climb	climb	1.0
climb	alaska	$7.5x10^{-5}$
climb	mount	$6.2x10^{-5}$

Table 3: Words that are similar to the query keywords in *Q*: "Climb Alaska" in the 3×10^{-5}-13% matrix

Word	Similar Word	Correlation Factor
alaska	alaska	1.0
alaska	borough	$3.0x10^{-6}$
alaska	pipeline	$3.7x10^{-6}$
alaska	yukon	$1.1x10^{-5}$
alaska	mountaineering	$1.4x10^{-5}$
alaska	climb	$7.5x10^{-5}$
alasta	mount	$4.5x10^{-5}$
climb	boulder	$3.5x10^{-6}$
climb	ceil	$7.9x10^{-6}$
climb	climb	1.0
climb	hike	$3.2x10^{-6}$
climb	ladder	$6.7x10^{-6}$
climb	min	$1.3x10^{-5}$
climb	rope	$5.2x10^{-6}$
climb	stair	$4.0x10^{-6}$
climb	steep	$4.7x10^{-6}$
climb	summit	$3.0x10^{-6}$
climb	mount	$6.2x10^{-5}$
climb	alaska	$7.5x10^{-5}$

Table 4: Words that are similar to the query keywords in *Q*: "Climb Alaska" in the 3×10^{-6}-13% matrix

Rank	Record #	Tags	Degree of Resemablance
1	7	alaska, climbing, mountaineering	2.0000166960
2	5	alaska, climbing, mountaineering	2.0000166960
3	10	alaska, climbing, hunting	2.0000154134
4	6	alaska, nature, history	1.0000756434
5	9	alaska, aviation, mountaneering	1.0000123297
6	8	alaska, mountain, memoir	1.0000121143
7	3	mountaineering, climbing, adventure	1.0000094294
...	...	...	...
15	4	photography, climbing, mountaneiring	1.0000094088
16	2	climbing, mountaineering, history	1.0000092849
17	1	alaska, photography, travel	1.0000003964

Table 5: Ten of the catalog records ranked for query *Q*: "Climb Alaska" using *Sim*

Rank	Record #	Tags	Degree of Resemablance
1	5	alaska, climbing, mountaineering	2.0
2	7	alaska, climbing, mountaineering	2.0
3	10	alaska, climbing, hunting	2.0
4	3	mountaineering, climbing, adventure	1.0000060541
5	9	alaska, aviation, mountaneering	1.0000045620
6	8	alaska, mountain, memoir	1.0000042120
7	2	climbing, mountaineering, history	1.0000014629
8	4	photography, climbing, mountaneiring	1.0000014579
9	1	alaska, photography, travel	1.0000013700
10	6	alaska, nature, history	1.0000001068

Table 6: Top-10 catalog records ranked for query *Q*: "Climb Alaska" using *LimitedSim*

Record #	Title
1	Alaska: Images of the Country
2	Climbing in North America
3	Facing the Extreme: One Woman's Story of True Courage, Death-defying Survival, and Her Quest for the Summit
4	Galen Rowell's Vision: the Art of Adventure Photography
5	In the Shadows of Denali
6	More Readings from One Man's Wilderness: the Journals of Richard L. Proenneke, 1974-1980
7	Mt. McKinley: the Pioneer Climbs
8	The Ascent of Denali: A Narrative of the First Complete Ascent of the Highest Peak in North America
9	Wager with the Wind: the Don Sheldon Story
10	Wilderness of Denali Explorations of a Hunter-Naturalist in Northern Alaska

Table 7: Titles of the ten ranked library catalog records shown in Tables 5 and 6

Table	Size in Memory	Prefix-string Index Size		
		3	5	8
tagid	227	162.9	172.9	189
correlation5en7	25	14.7	19	23
correlation3en5	4.5	2.7	3.5	3.8

Table 8: Size (in MB) of different indexed tables

Rank	HBLL Catalog	Our Enhanced Library System
1	The Road to Democracy in Iran	**My Traitor's Heart: A South African Exile Returns to Face His Country, His Tribe, and His Conscience**
2	**Old Wrongs, New Rights: Student Views of the New South Africa**	**Armed and Dangerous: My Undercover Struggle Against Apartheid (African Writers Series)**
3	The Spaces of the Modern City: Imaginaries, Politics, and Everyday Life	**Class, Race, and Inequality in South Africa**
4	Outsider Within: Reworking Anthropology in the Global Age	**The Invisible Line: Life and Photography of Ken Oosterbroek**
5	Seeking Mandela: Peacemaking Between Israelis and Palestinians	Love in Black and White: the Triumph of Love over Prejudice and Taboo
6	For Humanity: Reflections of a War Crimes Investigator	**Trade Unions and Democratization in South Africa**
7	**Class, Race, and Inequality in South Africa**	**Norms in International Relations: the Struggle Against Apartheid**
8	Rewriting Modernity: Studies in Black South African Literary History	**Internatinal Socialism 70 - Prisoner of the System**
9	**South Africa and the Logic of Regional Cooperation**	**The World that Was Ours**
10	**Race for Sanctions: African Americans Against Apartheid, 1946-1994**	**Race for Sanctions: African Americans Against Apartheid, 1946-1994**

Table 9: Top-10 library records (identified by their titles) retrieved for the query “Apartheid”