A Case Study of Latent Semantic Indexing

M.W. Berry, S.T. Dumais & A.T. Shippy

Computer Science Department

CS-95-271 January 1995

A Case Study of Latent Semantic Indexing[1]

Michael W. Berry
Department of Computer Science
107 Ayres Hall
University of Tennessee
Knoxville, TN 37996-1301
berry@cs.utk.edu

Susan T. Dumais
Bellcore
445 South Street
Room 2L-371
Morristown, NJ 07962-1910
std@bellcore.com

Amy T. Shippy
Department of Computer Science
107 Ayres Hall
University of Tennessee
Knoxville, TN 37996-1301
shippy@cs.utk.edu

January 6, 1995

[1]This research was supported by the National Science Foundation under grant Nos. NSF-CDA-9115428 and NSF-ASC-92-03004.

Abstract

In this report, a study and analysis of the effectiveness of the Latent Semantic Indexing Retrieval System (LSIRS) is presented. Using a Motif-based X-Windows application, LSIRS uses the truncated singular value decomposition (SVD) of the associated term-document matrices to perform document retrieval. The LSIRS user interface was initial prototype graphical user interface (GUI) of the recently developed XLSI application. The indexing and SVD software used to employ Latent Semantic Indexing (or LSI) was developed at Bellcore and the University of Tennessee.

Based on data collected from the usage of the system by graduate students and University of Tennessee library patrons, LSIRS is shown to be an effective and useful document retrieval system for both the inexperienced and advanced user. Suggestions for future system improvements are also described.

Contents

Chapter 1

Introduction

This chapter introduces the Latent Semantic Indexing Retrieval System (LSIRS), gives the motivation for performing a study of LSIRS, and discusses the anticipated outcomes of the study.

1.1 Overview of the Latent Semantic Indexing System

The Latent Semantic Indexing Retrieval System (LSIRS) [BAM93] is a Motif-based X-Windows application that performs document retrieval by searching with keywords and with documents entered as relevance feedback [SB90] by the user. The phrase *latent semantic* refers to the inherent underlying associations between words used to express a particular concept. LSIRS uses a latent semantic indexing technique, specifically singular value decomposition [GL89], in order to facilitate fuzzy searching based on patterns of word association. LSIRS uses singular value decomposition to model word usage patterns in user queries and compare them to usage patterns that occur within the documents of a database. This is in contrast to traditional document retrieval systems which attempt to match documents with queries through literal term matching.

Additional indexing and searching techniques employed by LSIRS include term weighting schemes and relevance feedback. Global term weightings de-emphasize the importance of terms that occur frequently across a database, and local term weightings emphasize the importance of a term within a particular document. Relevance feedback helps overcome difficulties users encounter in selecting optimal search terms by allowing the user to search using the text of relevant documents and thereby guide the system toward similar documents.

1.2 Motivation

LSIRS was developed as a joint project between the University of Tennessee and Bellcore. The indexing and retrieval software used to deploy Latent Semantic Indexing (LSI) was initially designed by Bellcore and later refined by students at the University of Tennessee. The LSIRS Motif-based graphical user interface (GUI) was developed by graduate students enrolled in the Department of Computer Science as an initial prototype of the XLSI application.

A study of the LSIRS system was performed [Shi95] in order to evaluate the usability of the Motif GUI and the effectiveness of the LSI for indexing and searching. To facilitate such a study, a graduate level seminar (Fall Semester 1992) in the Department of Computer Science at the University of Tennessee was offered. In this seminar, the students were required to observe users of the system and to submit reports on their observations. The 17 students from the Department of Computer Science and School of Library and Information Science who participated in the seminar each submitted one written report per week for a duration of eight weeks, for a total of 136 reports in all. This report comprises a formal analysis of the data gathered from the study (see also [Shi95]).

1.3 Anticipated Outcomes

The purpose of this study was to analyze the data contained in the weekly student reports in hopes of determining some measure of the usability of the Latent Semantic Indexing Retrieval System. The goal was to uncover trends, statistics, and commentaries from the users that would assist in the evaluation of the user interface and the search engine. It was hoped that the study would:

- Prove LSIRS to be an efficient means for locating documents within a collection and an improvement over performing a manual search of the same text.
- Show users' tendencies for using a particular method for reformulating a failed search, and determine if any strategy for refining a search is more productive than the others.
- Provide insight on the variance of search terms chosen by different individuals who are searching a collection for the same information.
- Determine if users become more skilled at using LSIRS as their experience with the system increases.

Chapter 2

Background Information

This chapter provides the reader with background information on the Latent Semantic Indexing Retrieval System (LSIRS). Section 2.1 gives an overview of the LSIRS search engine and Section 2.2 describes the LSIRS's user interface. In order to better understand the results of the study, it is essential that the reader have a basic familiarity with the material covered in the next two sections.

2.1 The LSIRS Search Engine

The LSIRS search engine was designed to overcome limitations and problems inherent in literal search engines. Specifically, latent semantic indexing techniques, relevance feedback [SB90] capability and term weighting schemes are utilized to obtain improvements over the performance of traditional retrieval systems that search for documents containing exact term matches.

2.1.1 Latent Semantic Indexing

Latent semantic indexing [DDF+90] is designed to overcome limitations inherent in traditional document indexing and retrieval systems that perform literal searches on query terms. Literal searching techniques overlook many documents because numerous words can be used to express the same idea. Also, since many words have multiple meanings, literal searches commonly yield documents which are unrelated to the subject matter of the query. In information retrieval literature, the concept of words having more than one meaning is referred to as *polysemy*, and the concept of more than one word having the same meaning is referred to as *synonymy*. Failure to address synonymy results in low *recall*, or the ratio of the number of pertinent documents retrieved to the number of pertinent documents contained in the database. Failure to address polysemy results in low *precision*, or the ratio of pertinent documents retrieved to the total number of documents retrieved by the query. LSI attempts to overcome these problems by using fuzzy searching techniques to locate documents that have the same conceptual meaning as the query.

The key idea behind latent semantic indexing is the deployment of a method for analyzing the semantic structure of the documents and queries to determine overall word usage patterns. Documents containing word usage patterns similar to those contained in the query are considered relevant to the query. LSIRS in particular utilizes linear algebra via the singular value decomposition [GL89], or SVD, of associated term-document matrices to approximate the underlying word structure of the documents. Conceptually the documents are plotted in an n-dimensional vector space, where n is the number of unique terms, or words, that appear across the database. Document similarity judgements are based solely on the dot product of, or cosine of the angle between, document vectors. Document vectors in close proximity of a query vector have a higher dot product with the query vector and are returned as the highest ranked documents judged to be similar.

The conceptual vector space is represented mathematically by constructing a sparse matrix of terms by documents defined by

$$A = [a_{ij}]\,, \tag{2.1}$$

where a_{ij} is the frequency of term i in document j. The construction of a terms by documents matrix is illustrated in Tables 2.1 and 2.2. Table 2.1 contains the text of three articles from the *Concise Columbia Encyclopedia* that discuss abortion and Supreme Court rulings on cases related to a woman's right to decide whether or not to bear a child. The indexed terms (or keywords) appear in boldface. Table 2.2 shows a terms by documents matrix constructed from the three articles. All terms that appear in more than one document are included in the matrix. This matrix was constructed by hand for the purpose of providing an illustration for the reader, however the actual matrices used by LSIRS are generated automatically by parsing software.

A truncated singular value decomposition (SVD) of matrix A is computed to approximate the original n-dimensional space of the terms by documents matrix in a space of k orthogonal dimensions, where k is substantially less than n. The discussion of the linear algebra techniques involved in computing the SVD of the terms by documents matrix is beyond the scope of this report. Readers interested in obtaining additional information on the singular value decompositions should consult [Ber92] and [GL89].

An obvious benefit of truncated SVD is that it reduces the complexity of the vector space, hence decreasing both the amount of disk space required to store the data and the time for real-time query analysis and data retrieval [Ber92]. Another benefit of the truncated SVD is that it attempts to diminish the influences of individual terms while preserving the primary term usage patterns. While each dimension in

Table 2.1: Concise Columbia Encyclopedia articles used as documents.

Document	Text
D1	**ABORTION** expulsion of the embryo or **fetus** before it is viable outside the uterus, i.e., before the 28th week after conception, in humans (see [reproduction]). Spontaneous **abortion**, or miscarriage, may be caused by the death of the **fetus** due to abnormality or disease or by trauma to the expectant mother. **Abortion** may also be induced, the **fetus** removed from the uterus by such methods as vacuum suction, dilation and curettage, intrauterine saline injection, the "**abortion** pill" (the drug RU486 in combination with another drug), and hysterectomy (surgical incision of the uterus). **Abortion** was long practiced as a form of [birth control] until pressure from the Roman Catholic Church and changing opinion led in the 19th cent. to the passage of strict **anti-abortion** laws. Attitudes toward **abortion** have become most liberal in the 20th cent. By the 1970s, **abortion** had been legalized in most European countries, the USSR, and Japan; in the U.S., according to a 1973 **Supreme Court** ruling (see [**roe** v. **wade**]), **abortions** are permitted during the first six months of **pregnancy**. **Abortion** remains a controversial issue in the U.S., however, and in 1977 congress barred the use of Medicaid funds for **abortion** except for therapeutic **reasons**.
D2	GRISWOLD V. CONNECTICUT **case** decided in 1965 by the U.S. [**supreme court**], establishing a **right** to privacy in striking down a Connecticut ban on the sale of contraceptives. The **Court**, through **Justice** [douglas], found a "zone of privacy" created by several amendments to the U.S. [**constitution**] guaranteeing against governmental **intrusion** in the homes and lives of citizens. The Griswold **decision** was important in later **cases**, such as [**row** v. **wade**].
D3	**ROE** V. **WADE** **case** decided in 1973, with a companion **case**, Doe v. Bolton, by the U.S. [**supreme court**]. **Justice** Blackmun, for the **Court**, ruled that states may not ban [**abortions**] in the first six months of **pregnancy**; that a **fetus** is not a "person" protected by the 14-th amendment to the U.S. [**constitution**]; and that the amendment protects a woman from state **intrusion** into her **decision** as to whether or not to bear a child. Blackmun asserted, however, that the **right** to an **abortion** is not absolute; After the first trimester the state may regulate that right for health **reasons**; after six months it may ban **abortions** except in **cases** in which the woman's health is in danger. The **Roe** **decision** led to attempts by **anti-abortion** groups to draft a constitutional amendment.

Table 2.2: Sample term by document matrix.

Term	D1	D2	D3
abortion	9	0	1
abortions	1	0	2
abortions	1	0	2
anti-abortion	1	0	1
cases	0	1	1
constitution	0	1	1
court	1	2	2
decision	0	1	2
fetus	3	0	1
intrusion	0	1	1
justice	0	1	1
pregnancy	1	0	1
reasons	1	0	1
right	0	1	1
roe	1	1	2
supreme	1	1	1
wade	1	1	1

the original n-space represented a single term, each derived dimension in the reduced k-space loosely represents a concept conveyed by a group of terms. As a result of the dimension reduction, documents with varying word usage patterns may be reduced to the same vector in the k-space. At the minimum, the truncated SVD causes the distance between document vectors due to differences in the word usage patterns of varying authors to diminish, so that similarity judgements are based on overall document content rather than the actual occurrence of words. However, it should be noted that some care and consideration must be given to selecting the proper number of dimensions k for the reduced matrix since too few dimensions will results in a loss of pertinent data [DDF+90].

User queries are processed by treating them as if they were small-scale *pseudo* documents. The query is represented as a vector sum of vectors corresponding to the individual words that comprise the query, and the query vector is plotted in the document vector space. Documents whose vectors are in the vicinity of the query vector are returned to the user as relevant documents. The closeness of a document vector to the query vector is determined by taking the dot product of the two vectors to obtain the cosine of the angle between them. The higher the cosine, the closer the vectors are to one another and the more potentially relevant the document is to the query.

In order to achieve a certain level of precision within the set of relevant documents returned to the user, a relevance threshold is applied. The threshold is typically set as the documents associated with the x closest document vectors, where x is some predetermined number, or as a cosine value which the dot product between the document and query vectors must equal or exceed. The documents are returned to the user in a rank-ordered list, sorted in descending order.

For the purpose of this study, two separate collections of data were analyzed using LSI. One of the collections consisted of six months of articles from the local newspaper, *The Knoxville News Sentinel*, and the second consisted of all of the articles from the *Concise Columbia Encyclopedia* (1989 Second Edition, on-line version). The dimension of the original terms by documents matrix generated from the newspaper articles was $40,140$ terms by $12,615$ documents. The dimension of the original terms by documents matrix generated from the encyclopedia articles was $29,670$ terms by $15,460$ documents. Documents parsed from both databases were comprised of the full article title and text. All words that appeared in a collection were included in the semantic analysis with the exceptions being words one letter in length, words occurring in a single document (*singletons*), and words included on a stop list of 1000 commonly-used words. No attempt was made to perform word stemming or to derive root form variants and include the resulting terms in the analysis. The truncated SVD was computed to reduce the dimensions for the newspaper and encyclopedia vector spaces to 310 and 236 dimensions, respectively.

2.1.2 Term Weighting

As is the case with many document retrieval systems, LSIRS utilizes global and local term weighting schemes to improve performance [Dum91]. Words are given a global weight to stress their importance across the total document collection, and a local weight to stress their importance within a particular document. Global weightings typically have an intended effect of diminishing the influence of words that occur frequently or in many of the documents.

Global and local weightings are applied prior to the singular value decomposition of the matrix A in Equation (2.1). The local weighting for term i in document j, $L(i,j)$, is applied to cell $a_{i,j}$, and the global weight for term i, $G(i)$ is applied across row i. As a result of the weightings, a new matrix

$$\tilde{A} = [\tilde{a}_{i,j}]$$

is derived from matrix A such that

$$\tilde{a}_{ij} = L(i,j) \times G(i).$$

In the terms by documents matrix shown in Table 2.2, local raw term frequency and no global weighting are used (i.e., $L(i,j) = a_{i,j}$ and $G(i) = 1$). The weighting schemes applied to the databases used in this study were raw term frequency local weighting and global entropy weighting

$$G(i) = 1 - \sum_j \frac{p_{ij}\log(p_{ij})}{\log(ndocs)},$$

where

$$p_{ij} = \frac{a_{ij}}{gf_i} \tag{2.2}$$

where gf_i is the number of times term i appears in the collection and $ndocs$ is the number of documents in the collection [Dum91].

2.1.3 Relevance Feedback

Relevance feedback helps guide users toward relevant documents by giving the system feedback as to which documents returned by a previous search are relevant to the initial query [SB90]. The system can then use the feedback to perform a subsequent search that will result in a list of documents with a higher precision and recall. LSIRS combines the text of the documents selected by the user to form a single *pseudo-document* and plots the vector sum of the terms found in the combined text in the document vector space. Users are unlikely to choose the best terms to express their queries. Relevance feedback allows them to use the words contained in relevant documents in a query which greatly increases the likelihood that the vector sum of the query will be plotted in the document vector space near relevant documents.

In research performed by Dumais, et. al [Dum91], usage of relevance feedback was found to greatly improve overall search performance. They discovered that queries composed from the highest ranked relevant document returned by the initial query gave an average overall improvement of 33% and queries composed of the three highest ranked relevant documents gave an average overall improvement of 67%. Their studies also found that the user typically must view only a small number of the documents returned by the initial search in order to locate a few relevant documents. On the average, the most relevant document was the top ranked document and the three most relevant documents were within the top seven ranked documents.

2.2 Description of the LSIRS User Interface

The following section describes the LSIRS user interface [BAM93]. The information covered in this section parallels the content of the LSIRS User's Manual that was made freely available to users of the system. The users were provided no other information in addition to what was contained in the manual.

2.2.1 Choosing a Book to Search

The initial LSIRS screen is shown in Figure 2.1. The screen contains an *Item* window displaying a list of the titles for available databases, or *books*. The user selects a book to search by using the mouse to point and click on an entry in the list. When a book has been selected, LSIRS highlights the list entry for the book and places the book title in the *Selection* subwindow. A book can be deselected by clicking on the *Cancel* pushbutton.

The user can click on the *Help* pushbutton at the bottom right of the screen to view an online help screen, or exit LSIRS by selecting *Quit* from the *File* pulldown menu on the menu bar above the *Item* window.

To proceed to the LSIRS *Search* screen and begin searching the selected book, the user must click on the *Ok* pushbutton. For the purpose of describing the user interface and illustrating the use of LSIRS, the remainder of this section assumes that the user selected the *Concise Columbia Encyclopedia* (CCE), and clicked on *OK*.

2.2.2 Performing a Search

Once a book has been selected, the *LSIRS Search* screen (Figure 2.2) appears and remains on the screen until the user chooses to exit.

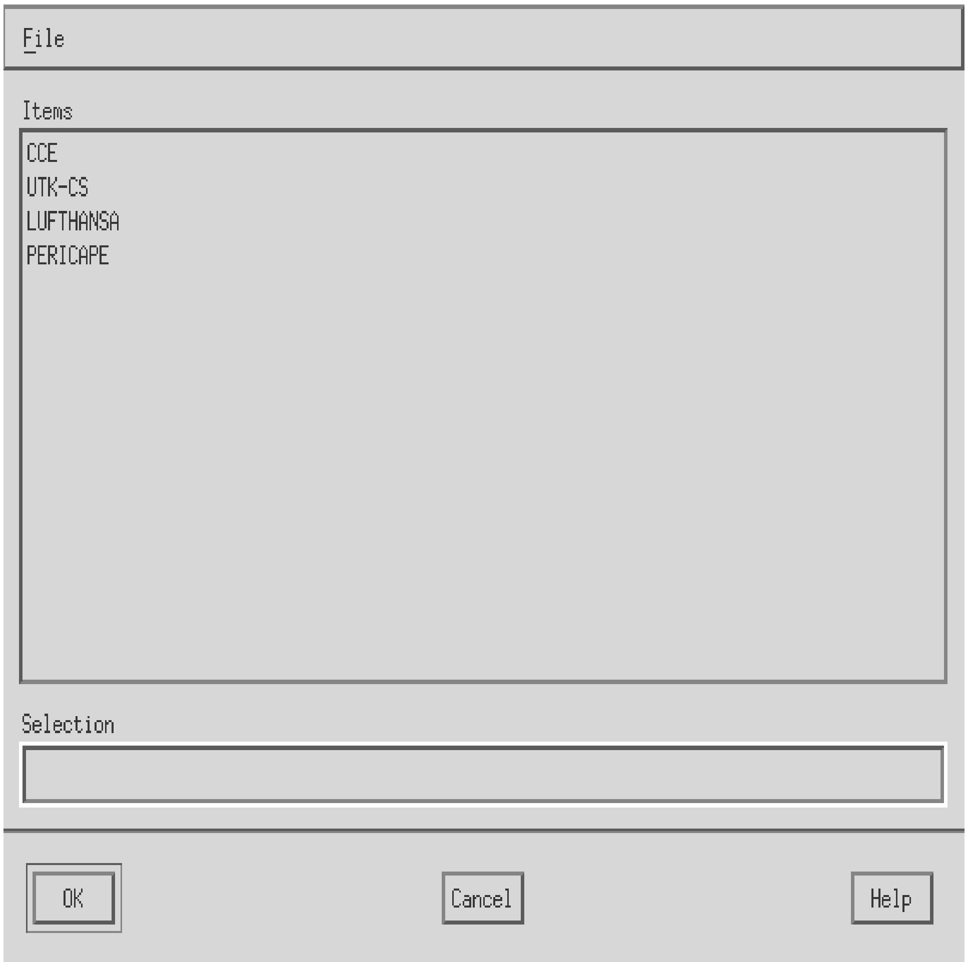


Figure 2.1: LSIRS startup screen.

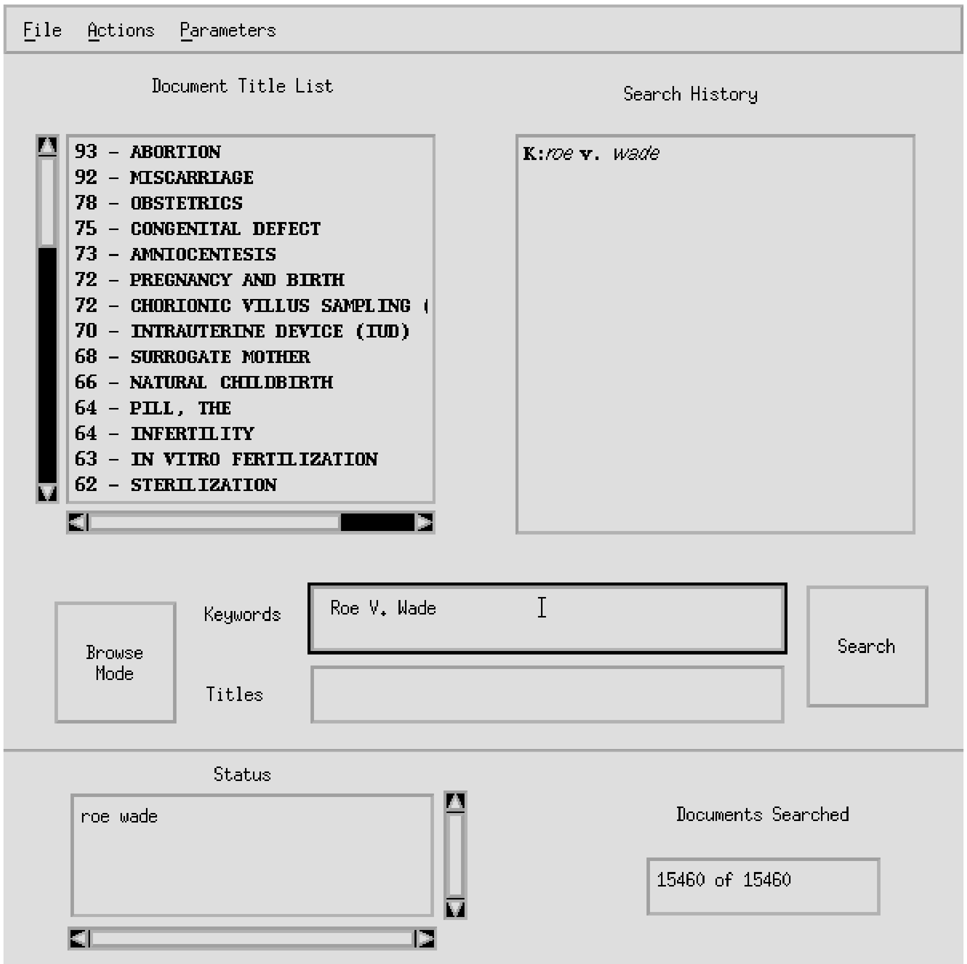


Figure 2.2: Composing a search.

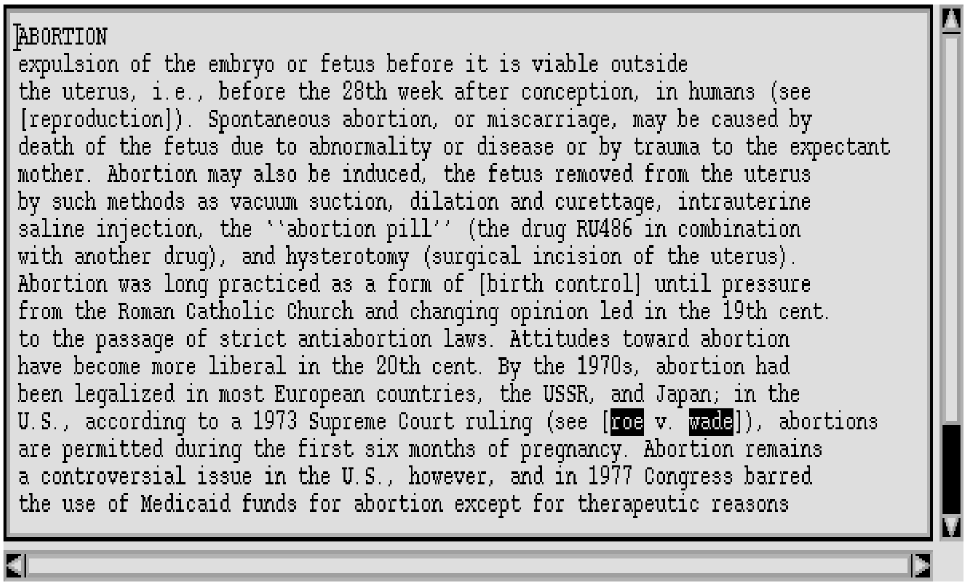
The titles of the documents in the book are initially sorted in descending alphabetical order, and the titles of the first 50 articles appear in the *Document Title List* window. The text from the article at the top of the *Document Title List* is immediately displayed in the *Document Text* window.

When the user first enters the *Search* window LSIRS is in *browse mode*, which means the user may browse through the documents in the *Document Title List* window by clicking on a document's title to trigger the display of its associated text in the *Document Text* window. To enter keywords for a query the user must first click on the *Mode* pushbutton to transition LSIRS to *search mode*. The label on the *Mode* pushbutton is toggled between *Browse Mode* and *Search Mode* to reflect the current mode of the system.

After LSIRS has been placed in *search mode*, the user may position the mouse pointer in the *Keywords* subwindow and enter search terms. If the user makes a typing error, the backspace key may be used to delete the mistake. If necessary, the mouse can be used to move the cursor to the character or character(s) to be deleted. After the search terms have been entered in the *Keywords* subwindow, the user triggers a search by clicking on the *Search* pushbutton.

When the search is initiated, LSIRS examines the search terms in the query for terms found in the original term-document matrix (i.e., indexed for the collection), calculates the sum of the corresponding term vectors, projects the result in the document vector space and takes the dot product of the query vector with each document vector in the space. If the number of documents in the book is large, the search may take a few seconds. LSIRS indicates its progress in computing the dot products of the query and document vectors by updating a counter in the *Documents Searched* window to reflect a running count of the number of documents searched out of the total number of documents in the book. After all of the dot products have been calculated LSIRS sorts the documents by their associated cosines (dot products) in descending order and returns a list of documents corresponding to the top 50 cosine values.

Upon completion of the search the *Documents Searched* window indicates that n of n documents have been searched, where n is the number of documents in the book. The query terms that were indexed and used in determining the vector sum of the query are displayed in the *Status* window. The titles of the 50-highest ranked documents appear in the *Document Title List* window, and the actual keywords that had been entered in the *Keywords* subwindow are logged in the *Search History* window preceded by a `K:` to denote that they were used in a keyword search. The text of the top document is shown in the *Document Text* window with the query terms highlighted (Figure 2.3). At this point, LSIRS has reverted back to *browse mode* to allow the user to browse through the titles in the *Document Title List* window by using the mouse to click on a document's title and thereby display its text in the *Document Text* window.



ABORTION
expulsion of the embryo or fetus before it is viable outside
the uterus, i.e., before the 28th week after conception, in humans (see
[reproduction]). Spontaneous abortion, or miscarriage, may be caused by
death of the fetus due to abnormality or disease or by trauma to the expectant
mother. Abortion may also be induced, the fetus removed from the uterus
by such methods as vacuum suction, dilation and curettage, intrauterine
saline injection, the ``abortion pill'' (the drug RU486 in combination
with another drug), and hysterotomy (surgical incision of the uterus).
Abortion was long practiced as a form of [birth control] until pressure
from the Roman Catholic Church and changing opinion led in the 19th cent.
to the passage of strict antiabortion laws. Attitudes toward abortion
have become more liberal in the 20th cent. By the 1970s, abortion had
been legalized in most European countries, the USSR, and Japan; in the
U.S., according to a 1973 Supreme Court ruling (see [roe v. wade]), abortions
are permitted during the first six months of pregnancy. Abortion remains
a controversial issue in the U.S., however, and in 1977 Congress barred
the use of Medicaid funds for abortion except for therapeutic reasons

Figure 2.3: Text of Highest Ranked Document.

2.2.3 Using Relevance Feedback Techniques

To perform relevance feedback queries using documents displayed in the *Document List* window, the user must first toggle LSIRS to *Search* mode by using the mouse to point and click on the *Mode* pushbutton Once LSIRS in *Search* mode, the user can select documents to use for relevance feedback by pointing and clicking on the titles of the desired documents in the *Document Title List* window. As the user selects documents LSIRS places their titles in the *Title* subwindow. The user may enter keywords in the *Keywords* subwindow to compose a *hybrid* query that consists of both keywords and documents. After the user has entered any keywords and selected the documents, the search is initiated by using the mouse to point and click on the *Search* pushbutton. LSIRS performs the search as described in Section 2.2.2.

The results of a relevance feedback search using the `Roe V. Wade` Concise Columbia Encyclopedia article (returned by the search performed in Figure 2.2) is shown in Figure 2.4. As with the case of the keyword search performed in Section 2.2.2, the titles of the top 50-ranked documents are displayed in the *Document Title List* window and the *Documents Searched* windows indicates that all documents in the book have been searched. The highest ranked document is `Roe V. Wade` with a relevance ranking of 100 (cosines were multiplied by 100 prior to display in *Document Title List* window), or cosine of 1.0, which reflects the fact that the document vector and the query vector are identical. Of course, one would expect the system to judge the contents of a document used for relevance feedback to be identical to the actual document.

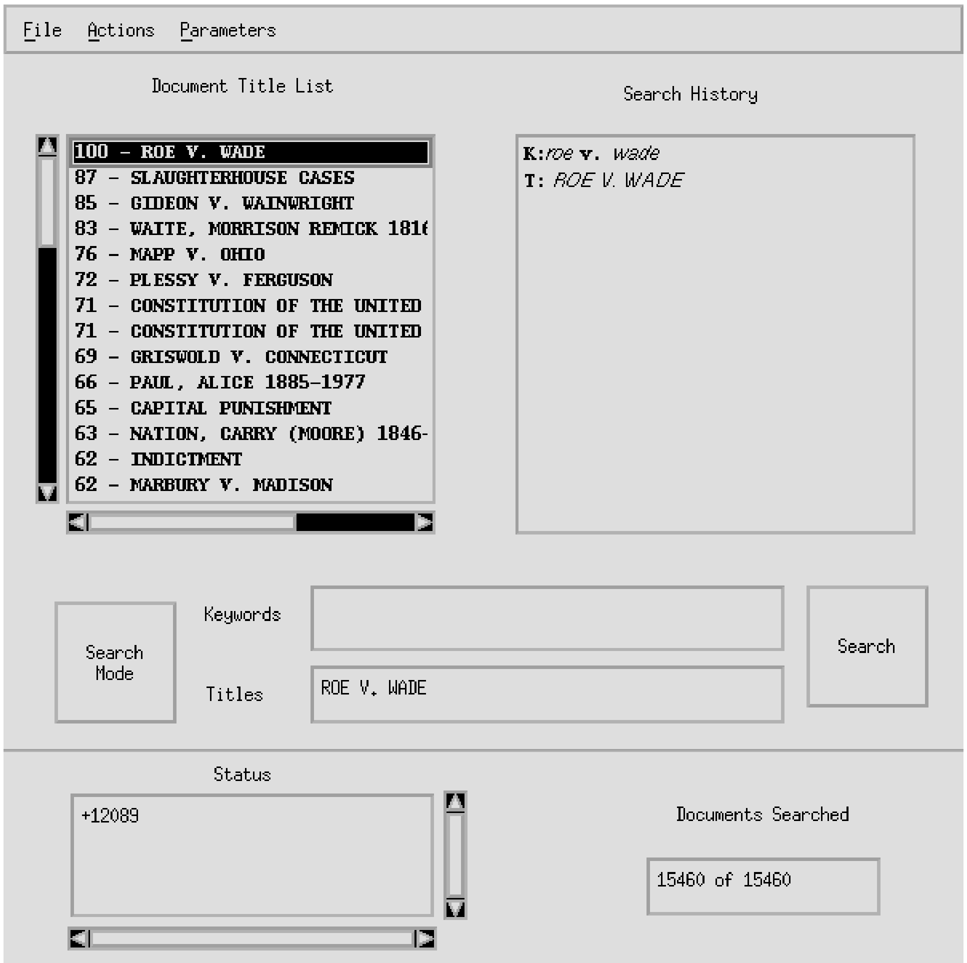
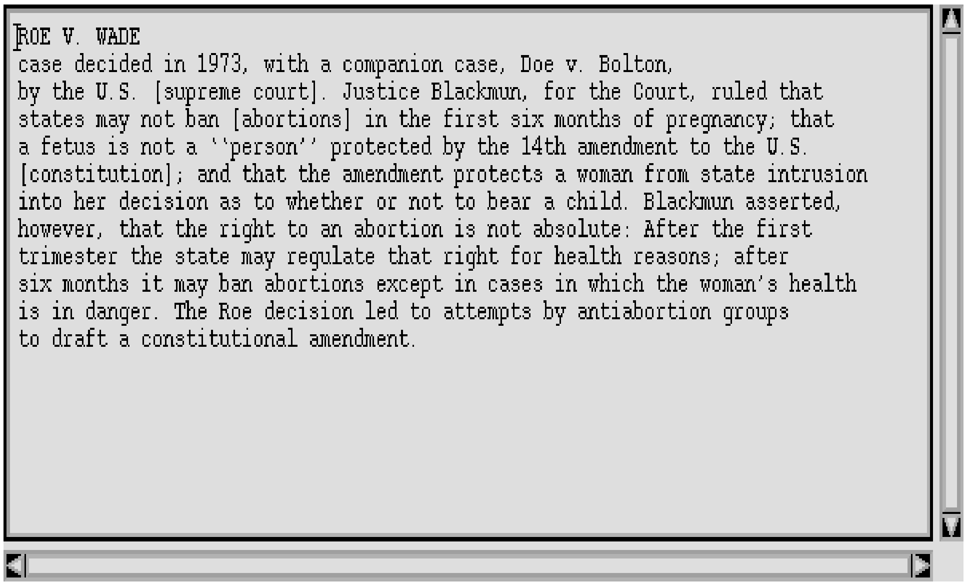


Figure 2.4: Performing relevance feedback.



ROE V. WADE
case decided in 1973, with a companion case, Doe v. Bolton,
by the U.S. [supreme court]. Justice Blackmun, for the Court, ruled that
states may not ban [abortions] in the first six months of pregnancy; that
a fetus is not a ``person'' protected by the 14th amendment to the U.S.
[constitution]; and that the amendment protects a woman from state intrusion
into her decision as to whether or not to bear a child. Blackmun asserted,
however, that the right to an abortion is not absolute: After the first
trimester the state may regulate that right for health reasons; after
six months it may ban abortions except in cases in which the woman's health
is in danger. The Roe decision led to attempts by antiabortion groups
to draft a constitutional amendment.

Figure 2.5: Text of document used for relevance feedback search.

The text of the highest-ranked document appears in the *Document Text* window (Figure 2.5). LSIRS doesn't highlight search terms taken from documents used for relevance feedback, so no words in the *Document Text* window are highlighted. Documents can be long and contain a number of unique terms. The purpose of highlighting search terms within documents returned by the search is to draw the user's attention to sections that might be particularly relevant to their search. If LSIRS highlighted terms from documents used in relevance feedback in many cases a large portion of the text in the *Document Text* window would be highlighted, which would defeat the purpose of highlighting terms. If the user had performed a hybrid query using both keywords and document text, the keywords entered by the user would be highlighted in the document text.

The unique identifier numbers for the documents used in relevance feedback are logged in the *Status* window. A line containing the document title(s) is added to the *Search History* window, and prefixed with a `T:` to indicate that the corresponding documents were used for relevance feedback. If the user had performed a hybrid query, the keywords used in the search would be added to the *Search History* window on a separate line with a prefix of `K:`.

When the relevance feedback search was completed, LSIRS once again placed itself back in *Browse* mode to allow the user to browse through the documents returned by the search. The 9-th highest document, with a relevancy ranking of 69 appears in Figure 2.6. By inspection of the text one can see that it is very similar to the `Roe V. Wade` document in that it discusses a Supreme Court ruling on a case concerning

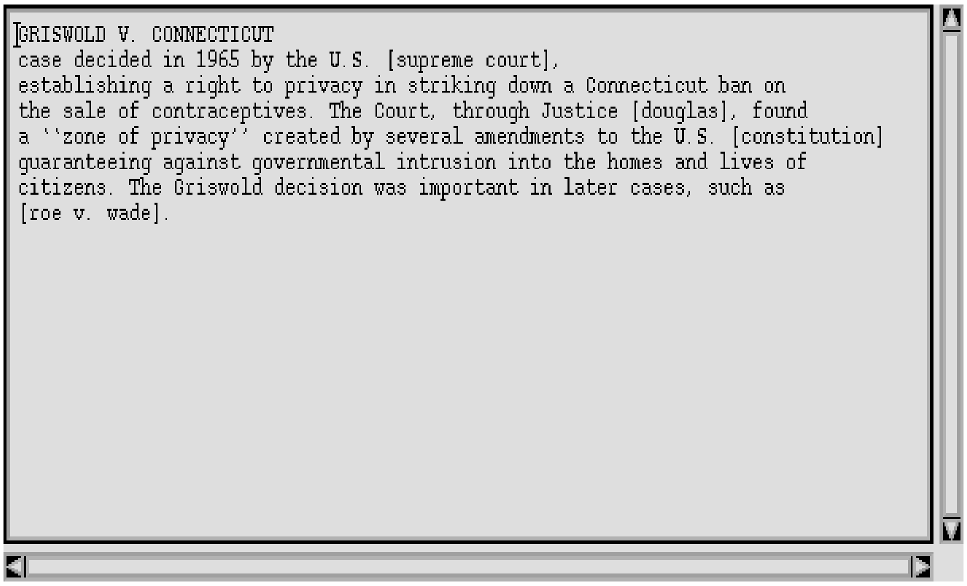


Figure 2.6: Text of document retrieved by relevance feedback search.

a woman's right to decide whether or not to bear a child, but uses different words.

2.2.4 Pulldown Menu Options

The menu bar above the LSIRS *Search* window contains several pulldown menus that give the user access to various functions that he or she might need during a typical LSIRS session. The user can print out or clear contents of the various display windows. The contents of the *Search History*, *Document Title List*, or *Document Text* windows can be printed by selecting the *Print History Text*, *Print Title Text* or *Print Document Text* options from the *File* pulldown menu (Figure 2.7) respectively. The *Keyword* and *Title* subwindows may be cleared by selecting the *Clear Keyword Search* or *Clear Title Search* options from the *Actions* pulldown menu.

The user can increase the number of titles displayed in the *Document Title List* windows by selecting the *Configure* option from the *Parameters* pulldown menu and entering a value in the *Maximum Documents* popup window. The new value is saved by choosing *Save* from the *Parameters* pulldown menu. The number of documents in the *Document Title List* window does not reflect the change until the next search is performed. A change made to the number of documents displayed is only in effect until the user modifies the value again or until the user selects the *Exit* option from the *File* pulldown menu to exit the LSIRS *Search* window.

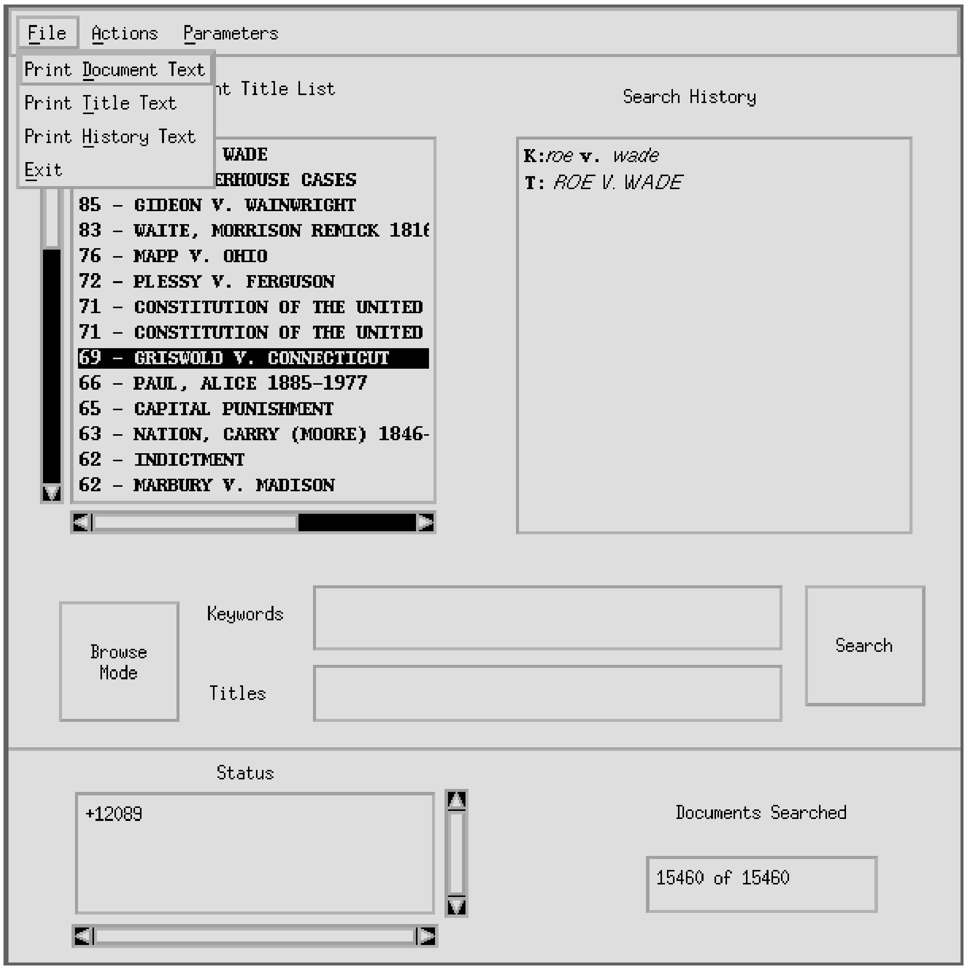


Figure 2.7: File pulldown menu.

Chapter 3

Case Study

This chapter provides details on how the study outlined in Chapter 1 was conducted. Section 3.1 discusses how the data was obtained by graduate students enrolled in a special topics course in which they were required to observe the usage of LSIRS by classmates and University of Tennessee library patrons, and submit reports summarizing their observations. Section 3.2 gives a brief overview of how the data contained in the student reports was standardized, entered into a database, and statistically analyzed.

3.1 Gathering the Data

The LSIRS software was installed on a Hewlett-Packard 9000-720 workstation and placed in the Reference Department of the John C. Hodges library on the University of Tennessee campus in Knoxville, Tennessee. An informal manual providing a brief but thorough overview of LSIRS was placed beside the workstation. Two databases that would be of general interest to librarians and patrons of the Reference Department were installed on the workstation. One of the databases contained back issues of the local newspaper, *The Knoxville News Sentinel*, from January 1, 1991 through June 30, 1991. The second database contained the complete text of the *Concise Columbia Encyclopedia, 1989, Second Edition* . (The Concise Columbia Encyclopedia was generously donated for use in the study by the Columbia University Press.)

The data collection was performed by seventeen Computer Science and Library Science graduate students who received three hours of graduate-level credit for participating in the study. Each student was assigned a weekly three hour time slot during which he or she observed patrons (and librarians) who used the system. If the patrons had difficulty using the system, the students were instructed to only answer patrons' questions on the user interface. Under no circumstances were the students to do any searches for the patrons since the purpose of the study was to examine the usability of the system. The students were also discouraged from explaining the details of latent semantic indexing. It was permissible, however, for the students to help the patrons in using the Hewlett Packard workstation, the mouse, and the printer.

The students submitted a weekly report on their observations and were graded on its clarity and content. In order to achieve some consistency in the information contained in the reports, a standard set of questions were answered for each patron observed. It was stressed to the students that the list should in no way be considered complete, but was intended to give them some guidance as to what should be covered in the reports. A list of these questions is included in the Appendix.

During periods when no librarians or patrons wished to use LSIRS, the students used the system themselves. While the students were encouraged to use LSIRS to search on topics of their own interest, they were also given weekly assignments that required them to use LSIRS to locate specific information. The students worked on the assignments in teams of two or three. Examples of queries from the assignments are provided in the Appendix.

The first three weekly assignments consisted of eight questions that could be answered from the *Concise Columbia Encyclopedia* on-line text. These questions were prepared by browsing through the LSIRS database and formulating questions that would require the students to successfully locate a particular article in order to determine the answer. Two teams competed to find the answer to each question, but one team was told to use LSIRS while the other was told to manually search for the answer in the hardcopy of the encyclopedia. Each team randomly selected two questions by drawing two slips of paper, one from the pool of on-line questions and the other from the pool of hardcopy questions. If a team drew a question for which any of the members of knew the answer, they selected a new question and returned their original selection to the pool so it could be assigned to another team.

For the remainder of the semester, except for the final week, the assignments were focused on the *Knoxville News Sentinel* articles. For the first assignment with the newspaper, the students were told to think of a cliche that would be likely to appear in the newspaper between January 1, 1991 to June 30th, 1991 and to search on that cliche. Some examples of the cliches used were *a thousand points of light* and *the mother of all battles*. After the initial search, the students were instructed to perform subsequent searches by formulating relevance feedback searches with documents returned by the previous search. The intent of this exercise was two-fold. First, the students were to see if the initial search returned documents that were indeed relevant to the true meaning of the cliche. Secondly, the students were to make note of the topics of any clusters of documents with a common theme that were returned by the searches.

For the second and third assignments with the *Knoxville News Sentinel*, each team was assigned a single question for which they were to find the answer by using LSIRS. No team was told to manually search the newspaper for the answer as had been the case with the *Concise Columbia Encyclopedia* assignments, since it would have been too difficult and time consuming for the students to browse through six months of back issues on microfiche. These questions were prepared by browsing through the LSIRS database and formulating questions that required students successfully locate a particular article, or set of articles, in order to determine the answer. The questions for these weeks were formulated around a common theme. Themes for these assignments included local environmental issues and the Persian Gulf War.

The final assignment with the *Knoxville News Sentinel* consisted of fifteen trivia questions on water. The questions were divided into three levels of difficulty, these being elementary school level, middle school level and high school level. The teams competed against one another on this assignment, the winner being the team which answered the largest number of questions. The questions were chosen at random with no prior knowledge as to whether they were actually in the LSIRS database or not.

The last assignment for the semester was for the students to write a two to three page description of any query on either database using LSIRS. The query could be a new one or one that had been performed before, but it had to be a good illustration of the usefulness of LSIRS. The description of the query was to be in the student's own words, could not be a joint team effort, and was to be thorough enough that it could serve as a tutorial for new users. The students were told to explicitly point out how to use any LSIRS features they utilized and to elaborate on any strategies they used in formulating their searches.

3.2 Analyzing the Data

Considerable time and effort was invested in putting the data contained in the student reports into a form that could be analyzed. The data needed to be transformed into a standardized format against which programs could be executed to generate statistics. A copy of FoxPro for Windows was readily available, so it was decided that the first step would be to enter the data into a relational FoxPro database. The students had submitted their homework assignments and observation reports via electronic mail, so the data was available in electronic form. However, due to the fact that the students had not been given a standardized format for keying in their reports, there was no way to automate extrapolating the raw data from their electronic mail messages and placing them into the proper database fields. The only option was to read through the data files and manually enter the data into the database.

As the data was entered into the database, it became obvious that some pertinent information was missing. Although the students were given a set of standardized questions, there was still a great deal of inconsistency in the level of detail contained in their reports. Early in the semester the students had been asked to provide the rank and cosine of the document that contained the answer to their homework assignment.

Some of the students consistently provided the rank and cosine information for the duration of the semester on both homework assignments and ad hoc searches they performed, but many did not. In addition to the rank and cosine being missing for many of the student performed queries, neither of the items were provided for the searches performed by library patrons. In fairness to the students, it would have been awkward for them to have obtained this type of information on the patron observations. The students were instructed to explain that they were participating in a study of LSIRS and if the patron would mind if he or she was observed. The students discovered early in the semester that a majority of the library patrons did not want to be observed. In these situations the best the student could do was observe the patron's interactions with LSIRS from a distance and examine the search history after the patron left. However, the keywords used in all but a few of the searches were known, and quite often the titles of documents used in relevance feedback were provided, therefore the information necessary to recreate the searches was available. In order to fill in the rank and cosine data missing from the student's reports, it was necessary to redo the searches. As many of the searches as possible were reprocessed via a C-shell script using a command line interface to LSIRS, but a large number of the searches had to be performed manually via the LSIRS user interface.

After the data was installed in the FoxPro database, a file containing a formatted, labeled listing of the data was produced. The file was uploaded from the PC environment to a Sun Workstation where PERL [WS90] scripts were written to scan the formatted file and calculate statistics on the data. The scripts examined the data from various angles, comparing on-line versus manual searches, calculating the success and frequency of use for modification strategies, examining word usage trends for queries that were performed by multiple users, and looking at usage statistics by week for queries performed by the students. The statistics generated by the scripts are discussed in detail in the next chapter.

Chapter 4

Results of the Data Analysis

This chapter discusses the results of the analysis performed on the data collected by the students. Section 4.1 establishes the definitions for the terminology used in describing the findings of the study. The remaining sections convey the findings of the study. Statistics on search success and failure rates, cases where the same information was searched by multiple users and trends in student usage techniques as the semester progressed are presented.

4.1 Terminology

Before the results of the study are presented, several terms used to describe the results should first be defined. In the discussions that follow, the term *query* is used to denote a particular user's or student team's attempt to locate an answer to a specific question or information on a certain topic. A query consists of a series of related *searches*, including an initial search and one or more subsequent searches in which the user or team tries to modify the initial search in order to improve upon the search outcome. Searches are categorized based on when they occurred within the series of searches for a query. The initial search is referred to as the $1/n$ search, the second search as the $2/n$ search, the third search as the $3/n$ search, etcetera. The final search in a series is referred to as the n/n search, where n denotes the total number of searches in the series.

In general, when a user utilizes a particular methodology to modify an earlier search in order to improve upon its results, he or she is considered to be using a *modification strategy.* Users' attempts to formulate a new search by revising a previous search are categorized into three major categories: keyword modifications, relevance feedback modifications and hybrid modifications. Searches classified as *keyword modifications* are those which involve only a change in the keywords used. Keyword modifications fall into one of three subclasses based on whether the search has more keywords than the initial search, fewer keywords than the initial search and or the same number of keywords as did the initial search. *Relevance feedback modifications* include searches which use one or more documents from the initial search and no accompanying keywords. *Hybrid modifications* are searches which use relevance feedback along with one or more keywords.

The results of searches performed by the graduate students who assisted in conducting the study are examined separately from those performed by librarians and library patrons. The graduate students received training on how to use the system in order to prepare them for their participation, and therefore had a better understanding of how the system worked than would the casual user. This segregation of the data is interesting because it allows the success rate and modification techniques of the experienced student users to be contrasted with those of the lesser-experienced patrons and librarians, and it permits the examination of changes in modification technique usage and success rates among the student users as they gained experience over time. When the data is broken down by user classes, the graduate students enrolled in the course will be referred to as the *students*, and users not enrolled in the course will be referred to as *patrons*. The term *user*, or *users*, will denote students and patrons collectively.

4.2 Analysis of Search Success and Failures

The data was analyzed to determine the success and failure rate of the system. The overall success rate, the success of the initial search versus subsequent modification attempts, and the success of the various modification classes were examined. Additionally, the success of manual versus on-line queries was examined for cases where students searched for answers to questions in the Concise Columbia Encyclopedia both manually and with LSIRS.

4.2.1 Overall Query Success

The overall query success rates are shown in Figure 4.1. Across the entire user community, 72% of the queries performed were ultimately successful. In most cases if a query was unsuccessful the user felt confident that the information he or she was searching for was not in the database.

Queries performed by the students were successful 75% of the time, while queries performed by patrons were successful only 57% of the time. It is not surprising that the queries success rate for the students was higher than the success rate for the patrons for two reasons. First, the students had received some training in the usage of LSIRS prior to their first attempt to use the system. Secondly, the students were repeat users over an eight week time period, so one would expect the students to naturally become more adept at searching with LSIRS as their experience with the system increased.

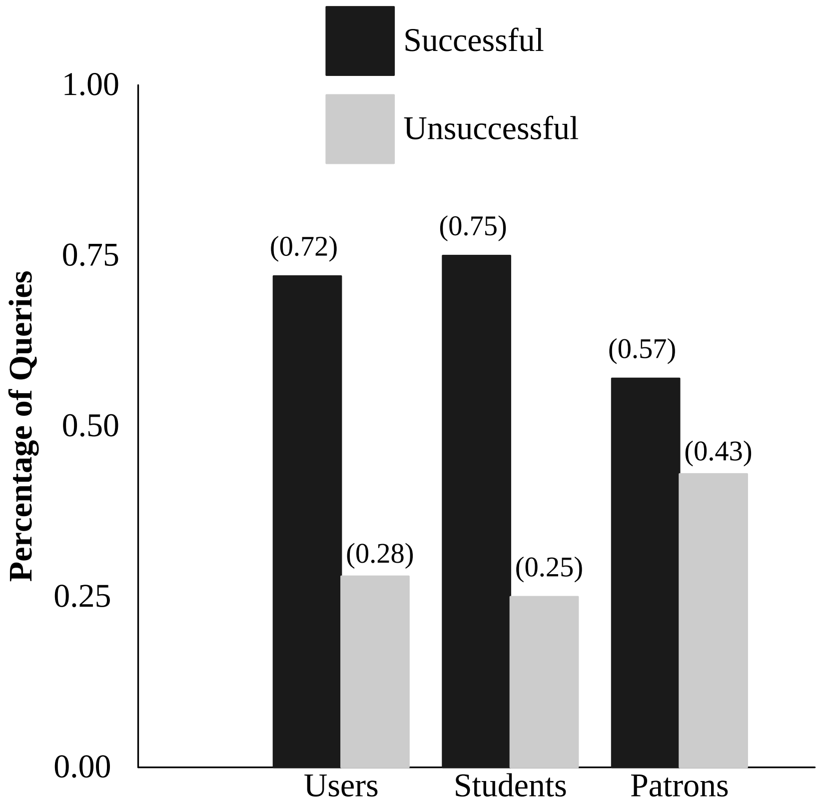


Figure 4.1: Overall success of queries.

4.2.2 Success of Manual Versus On-line Searches

Figure 4.2 shows success rates of on-line versus off-line queries in cases where the students were assigned homework which required them to perform the same query both on-line using LSIRS and off-line by manually browsing through the Concise Columbia Encyclopedia. Queries performed using LSIRS were 90% successful, while only 68% of the manual searches were successful. The success rate given here for on-line queries is higher than the overall query success rate cited in Section 4.2.1 because the queries assigned to the students were derived from articles in the Concise Columbia Encyclopedia.

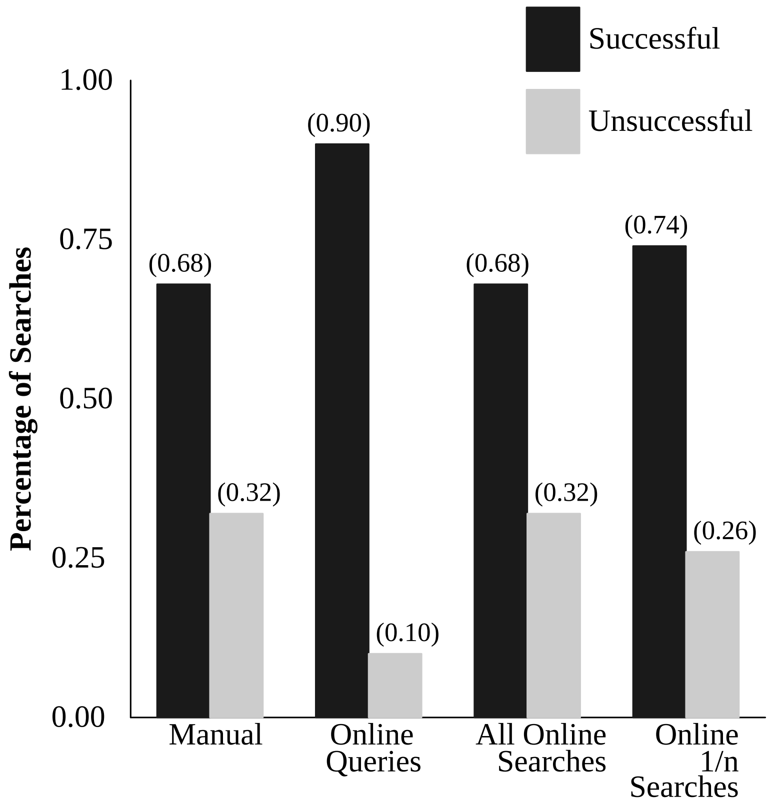


Figure 4.2: Success of manual versus on-line searching.

Students performing on-line queries had a 74% likelihood of locating the answer on their first attempt, but if the answer was not found on the first search it was worthwhile to make subsequent search attempts since 52% of the $1^+/n$ searches were successful.

Comments from students indicate that manual queries were unsuccessful in more than one instance because the answer was the actual title of the article that contained it, therefore the student would need to know the answer in order to locate the answer. (One example of such a query is *Who would have become president if Andrew Johnson had been impeached*. The answer to this particular query, *Benjamin Franklin Wade* is located in the article entitled *Benjamin Franklin Wade*.) Several students reported that after having spent a great deal of time searching the Concise Columbia Encyclopedia with no success, they declared the manual search attempt a failure but, since they were curious to learn the answer, they opted to use LSIRS and were then able to quickly locate the answer.

The students also reported that the on-line queries were unsuccessful when a key search term occurred in a single article because terms appearing in only one article were discarded from the original terms by documents matrix derived from the Concise Columbia Encyclopedia text (see Section 2.1.1). For example, if the user was attempting to find the meaning of the term *syzygy* without any knowledge of the context the term is usually used in, a likely beginning search might be a single keyword *syzygy*. The term *syzygy* only appears in the article *syzygy*, therefore it is not included (by default) in the terms by document matrix and will not be included in the query vector calculations.

4.2.3 Success of Initial Search and Subsequent Modification Attempts

Figure 4.3 illustrates the success rates of initial searches and subsequent attempts to refine an initial search. Only the first six search attempts for any series of searches were examined because there was not an adequate number of series that consisted of seven or more searches.

The user was most likely to find the information they were seeking on the first search attempt. The initial search was successful 63% of the time. The success rates of the subsequent searches decreased significantly, with the $6/n$ search successful 17% of the time. The biggest drop in success rate occurred between the first and the second search, where there was a decrease of 30%. Since the overall success rate of the queries was 72%, the answer to the successful queries was located on the first search 87% of the time. It is interesting to note that the mean rank of the document containing the answer was fairly constant regardless of which search it was located on. The mean rank was 19.88 for $1/n$ searches, 22.54 for $1^+/n$ searches, and 23.76 for n/n searches.

The success rate for subsequent searches for the on-line queries in student homework assignments performed against the Concise Columbia Encyclopedia (Section 4.2.2) was 52%, significantly higher than the overall rate of success for subsequent searches. This discrepancy may be due to the fact that the queries associated with the student assignments typically required the user to search for an answer to specific question. The user might not find the answer on the first search attempt, but could then perform relevance feedback using a document returned by the first search to steer the system to the article that contains the answer. However, the calculations for the overall success of subsequent searches include queries in which the user is merely searching for any information on a particular subject, and in such cases it is highly probable that LSIRS would return relevant information on the first search attempt.

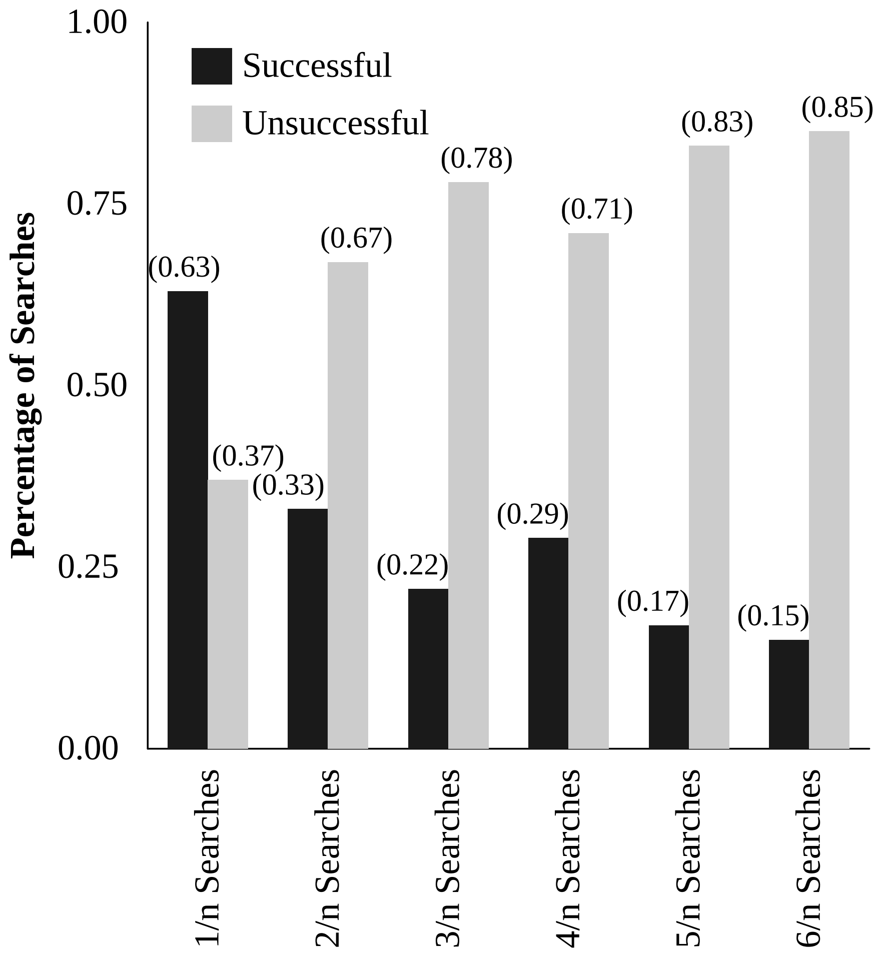


Figure 4.3: Success of x/n ($1 \leq x \leq n$) searches.

4.2.4 Success of Modification Strategies

Analysis of the success rates of the various modification techniques showed keyword modifications to be the less successful than either hybrid searches or relevance feedback alone (Figure 4.4). Relevance feedback searches were successful 31% of the time. The hybrid modifications were the most effective, successful approximately 41% of the time. Keyword modifications failed to locate pertinent information in 81% of the cases.

Keyword modification was most effective when the number of keywords used in the search was decreased (Figure 4.5), but even then the success rate was only 25%. The students reported that they found keyword searching to work best when they made an effort to select a few terms that were fairly unique to the intended target subject of their query therefore unlikely to occur in other contexts. One student commented that if the term *rock* was used to search for material on rock music, many of the articles returned by the search would be related to geology. This observation indicates that LSIRS is still somewhat effected by polysemy.

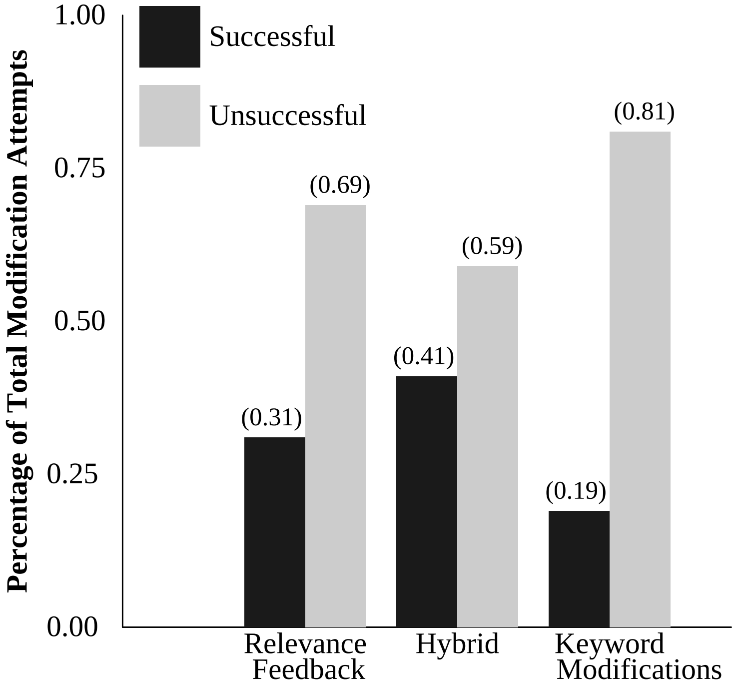


Figure 4.4: Success of modifications by category.

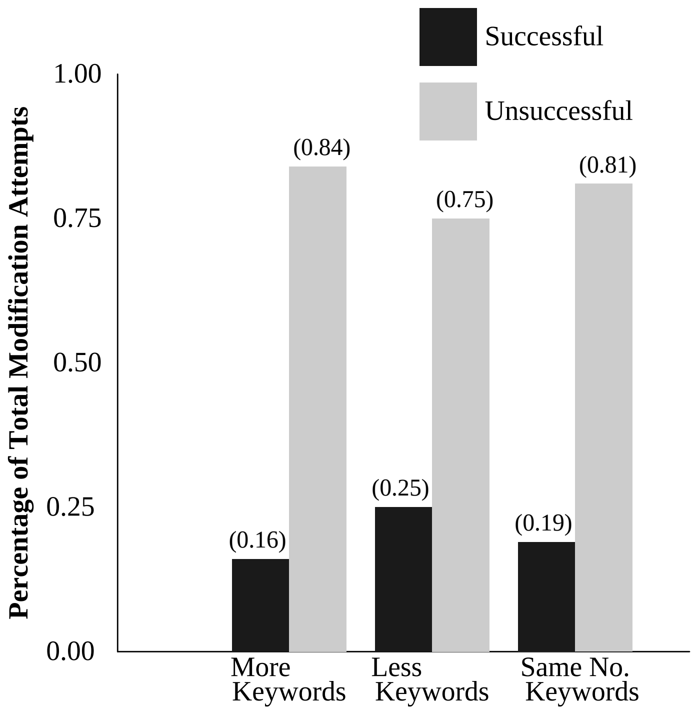


Figure 4.5: Success of keyword modifications by subcategory.

4.2.5 Usage of Modification Strategies

The overall user tendency to select a particular strategy for modifying the initial search was tabulated along with the tendency to use more, less or the same number of keywords as the original search when a keyword modification was performed. Users were most likely to refine the keywords used for a subsequent search (55%) and significantly less likely to use relevance feedback (31%) or hybrid searches (14%). A preference for keyword searches is not surprising, since the user would be accustomed to using traditional document retrieval systems that permit only keyword searches. Ironically, users chose to increase the number of keywords most often (28% of total modification attempts) in spite of the fact that this modification strategy was the least effective of the modification strategies.

Modification strategy usage for the student and patron subgroups is shown in Figure 4.6. The students and patrons used keyword modification techniques in approximately 51% and 65% of subsequent searches, respectively. Both groups tended to increase the number of keywords used when performing a keyword modification, and both formed relevance feedback searches a little less than 1/3 of the time.

The students used hybrid modification techniques more often than the librarians and library patrons. The students commented that hybrid searches had an advantage over relevance feedback alone because keywords used in the hybrid searches were highlighted in the document text window and this made it easier for them to spot the answer to a query when browsing through the documents in the *Document Title List* window. The students seemed to feel that if the keywords they were looking for were not brought to their attention by highlighting, they would overlook relevant documents, especially if a document was rather lengthy and the answer to their query was located somewhere in the middle of the text.

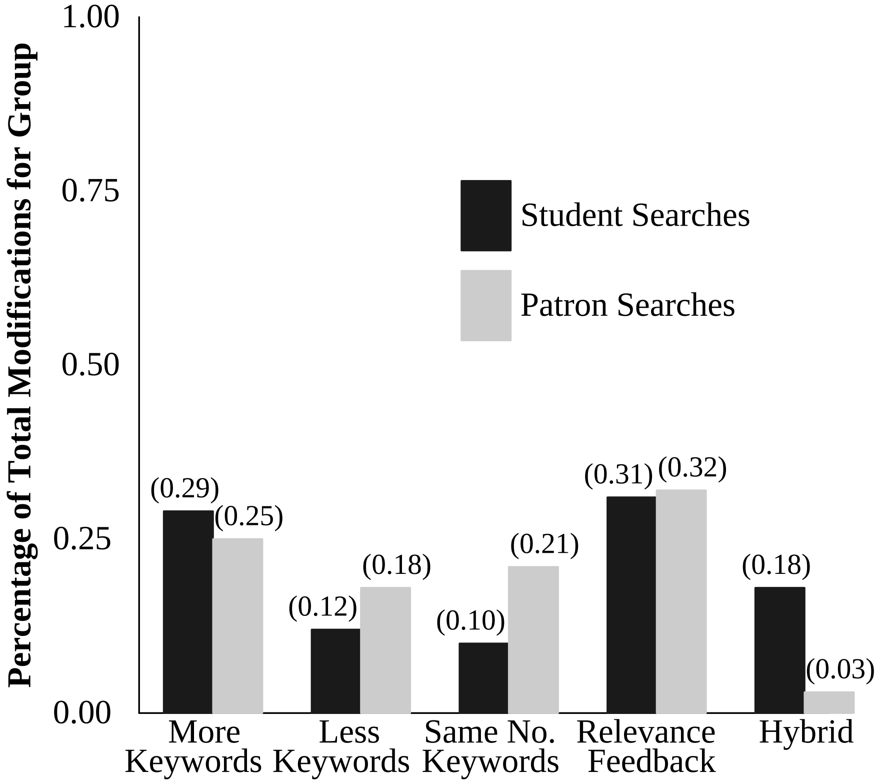


Figure 4.6: Student and patron usage of modification techniques.

4.3 Analysis of Queries Performed by Multiple Users

The initial ($1/n$) searches for queries performed by more than two users were analyzed to determine the variance in the search terms chosen. Identical $1/n$ searches occurred for 22% of the queries when the order of the keywords is considered, and in 24% of the queries if order of the keywords is ignored. When a query did have $1/n$ searches that contained duplicate terms, regardless of order, from 20% to 100% of the initial search attempts for the query were identical.

The first terms of the keyword phrases used in the $1/n$ searches were compared to obtain the frequency of identical first terms. Specifically, the percentage by query of $1/n$ searches which began with a word that was chosen as a first term by multiple users was recorded. Overlapping first words occurred across all of the queries, with 95% of the queries having a first word in common for at least 50% of their $1/n$ searches. 29% of the queries had duplicate first terms occurring in 100% of the first search attempts. Nineteen percent of the queries had two or more words that appeared as a first term more than once.

The frequency of overlapping terms regardless of their position in the initial search phrase is worth noting. For 97% of the queries, 60% or more of the terms were used by multiple users. Forty-three percent of the queries had overlapping terms 90% of the time, and 20% of the queries had 100% overlapping terms.

The reason for the fairly frequent occurrence of duplicate queries and a high rate of overlapping search terms is likely due to the fact that most of the queries performed by multiple users were part of student homework assignments. The queries were presented to the students in the form of a question, e.g., *What percentage of the human body is water?*. When composing the first search the students were naturally inclined to read through the question, selecting terms as they scanned from left to right. Hence, it is not surprising that often the students selected the same terms.

4.4 Analysis of Student Searches by Week

The student queries were broken down according to the week of the semester in which they were performed then examined to determine if either student preference for modification techniques changed or the success rate increased as the students became more experienced with using the system. Figure 4.7 shows the success of student performed queries by week. The success rate was fairly consistent during the weeks when the answers to the queries given in student homework assignments were known to be present in one of the databases. There was a slight decrease in the success rate during the second and third weeks, and a slight increase during the sixth and eighth weeks. The success rates for the fourth week, when the students were told to use cliches as their search phrases, and the seventh week, when the assigned queries were randomly chosen with no guarantee that the answers were actually in either database, were significantly lower. Similar success rates occur when the statistics are calculated on the $1/n$ searches for each query (Figure 4.8). These statistics show that LSIRS performs consistently well for both novice and experienced user.

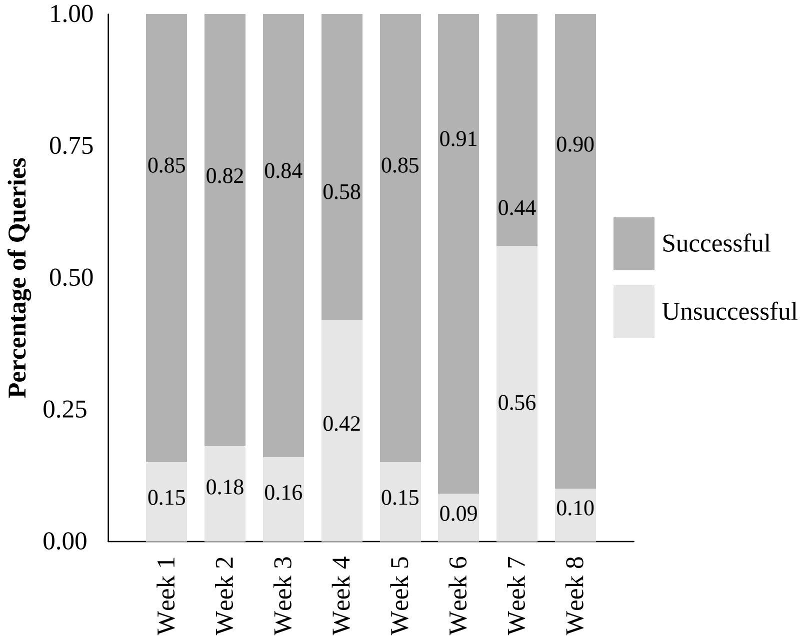


Figure 4.7: Success of student-performed queries by week. Weeks 4 and 7 reflect queries in which target information was not known to exist in the database.

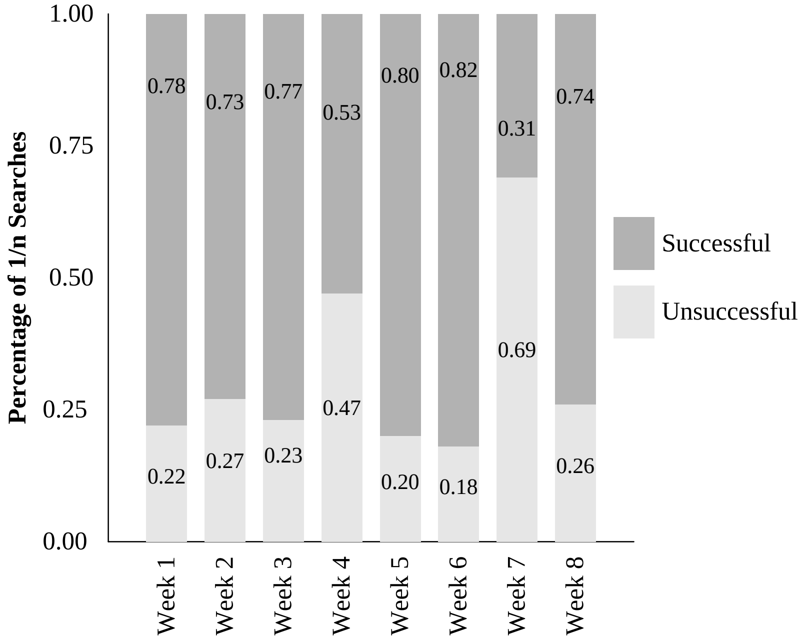


Figure 4.8: Success of $1/n$ ($where\ n > 0$) searches for student-performed queries by week. Weeks 4 and 7 reflect queries in which target information was not known to exist in the database.

Figures 4.9 and 4.10 illustrate the student tendency to select particular modification techniques as the semester progressed. The data shows no evidence of a relationship between user preference for modification techniques and the user's experience with the system. Several hypotheses can be formed to explain why certain modification techniques were selected during the fourth, seventh and eighth weeks. There was a high percentage of relevance feedback modifications during week 4 because the students were instructed to perform relevance feedback on the documents returned by the cliches used as their initial searches. A high percentage of keyword modifications occurred in week 7 since the students were unable to find information relevant to their query and therefore did not locate any documents that were candidates for relevance feedback. Also, it is interesting to note that in the eight week, when the students were told to compose a query and write a detailed narrative for a user's guide describing how they used LSIRS to perform the query, the students chose to perform hybrid searches most often.

Overall, the students were just as likely to use any of the modification techniques during the early weeks of the semester as in the final weeks. Since the students were lectured on the usage of LSIRS prior to performing their first query, they were somewhat better educated on its use than the typical library patron who simply happened upon the system and attempted to use it. However, the statistics do suggest that with minimal training a user can successfully choose appropriate modification techniques to navigate LSIRS towards the answer to a query.

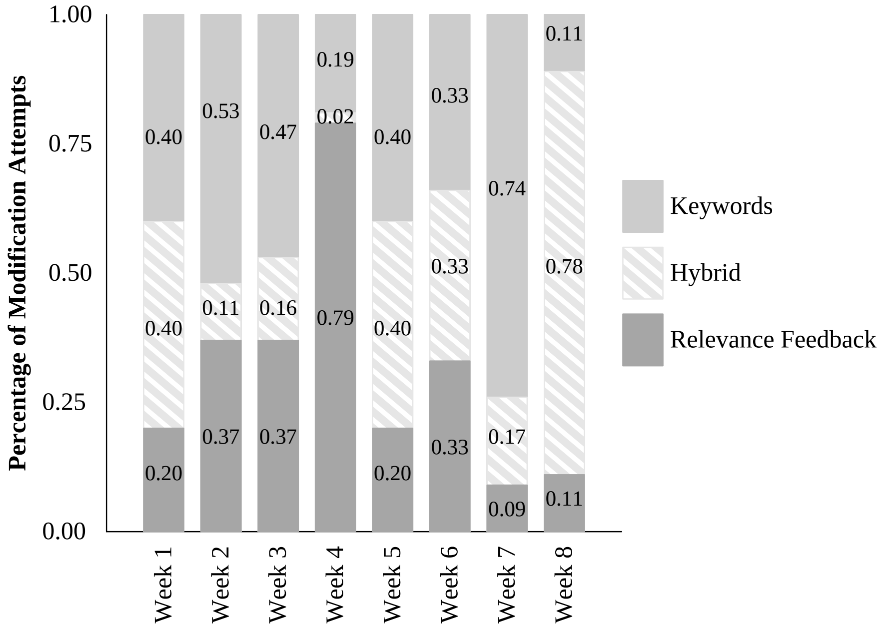


Figure 4.9: Student usage of modification techniques by week.

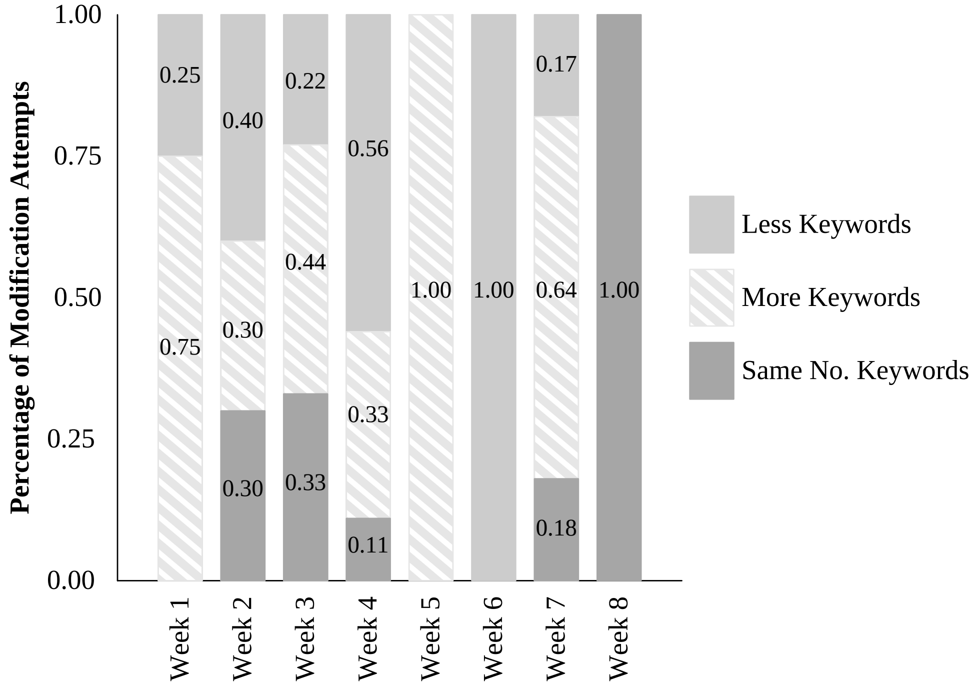


Figure 4.10: Student usage of keyword modification techniques by week.

Chapter 5

Conclusions

This chapter summarizes the results of the study, cites possible improvements and enhancements to the system, and suggests how future studies on LSIRS might be conducted.

5.1 Summary of Study Results

Three of the four anticipated outcomes of the study were met. LSIRS was shown to be an effective means of information retrieval for the novice user, and that, with minimal training, the user can quickly become proficient in using the system. Moreover, on-line searching via LSIRS was shown to be more effective than attempting to locate information by manually searching through the text. The study also provided insight on how often initial search attempts failed, how users formulated subsequent searches, and on whether any of the modification strategies were more productive than the others. Over half of the time the answer to a query was found on the initial search attempt. When the initial attempt did fail the users showed no tendencies toward a particular modification technique, but appeared to chose what they thought to be the best strategy for the given situation. However, when the success rates for the modification techniques were compared, hybrid searches proved to be the most successful.

The study failed to provide useful data on the variance of terms chosen by users who performed the same query. Although a high number of overlapping terms occurred, it is likely that this is due to the fact that the students were biased toward the selection of words used to communicate the assigned query to them. One would expect a greater variance in terms if the students had composed the queries themselves. Instructing the students to select search terms that do not appear in the assigned query might help in overcoming this bias.

5.2 Future Work

Improvements to Database and Document Retrieval Engine

Suggested possible improvements to the database include indexing singletons, applying a stronger global weighting scheme to further dampen the effects of frequently occurring terms, and assigning a heavier local weight to terms that appear in the document title. Indexing terms that appear in only one document would achieve better success rates for data collections that cover a broad range of unrelated topics and are likely to have a higher proportion of singletons. A global weighting scheme that further diminishes the effects of commonly used terms should be used to counteract LSIRS's tendency to be misdirected by terms used loosely within a wide variety of contexts. Terms in a document title should be given a heavier local weight since they play an important role in conveying the subject matter of the document.

The incorporation of the ability to transition the search engine to boolean search mode to perform literal term matching should be considered. Also, the addition of a phrase searching capability that permits the user to indicate when search terms must appear adjacent to each other in the document text would be beneficial.

Enhancements to User Interface

LSIRS restricts searching to a single database at a time and forces the user to return to the *Startup* screen to switch from one database to another. The interface should be redesigned to allow the user to select multiple databases for simultaneous searching, and permit the selection of the databases from the LSIRS *Search* screen.

The user should be able to redisplay the results of a previous search in the *Search* screen by selecting the search from the *Search History* list. Currently the user can reconstruct a search by dragging keywords and titles from the *Search History* list to the *Keywords* or *Titles* subwindows, but must reiterate the search in order to recall its results.

The state of the *Search* screen is lost when the user exits to the *Startup* screen. The ability to save the state of the *Search* screen so that it may be restored during a future session would be a useful feature.

Suggestions on How Future Studies Should be Conducted

The subjects in this study were from two distinct groups of users. The patrons were users by happenstance, they chose to use the system because the databases contained information that could benefit them. The students actively chose to participate in the study by enrolling in the course. Valuable data can be gained from observing both of these groups, but the collection of data from each group must be approached in different manners.

The task of collecting data on users by happenstance might be better facilitated if there was an automated means of logging the searches performed and the cosines, ranks and titles of documents browsed by the users. A major obstacle encountered in collecting the data for this study was finding patrons who were willing to be observed. Many of the patrons were not receptive to having a student observe their interaction with the system, and for this reason sufficient data was not obtained on a number of the patrons' searches. The logging of the patron's activities would have helped enable the collection of this missing data.

The information obtained from active participants in the study can be much more detailed than that obtained from passive participants. In future studies, active participants should be again be given a list of questions to answer about the queries they perform, but the questions should be as specific as possible. For example, the participants should be asked to record the keywords and relevance feedback titles used in all searches, the rank, cosine, title and identifier of the document containing the answer to the query, and the number of documents they browse before locating the answer. They should be given a strict format for submitting the data so that it can be loaded into a database with minimal manual intervention.

Acknowledgements

The authors would like to thank James Raimes, Editorial Director of the Reference Division at Columbia University Press, for providing an on-line version of the 1989 (Second Edition) Concise Columbia Encyclopedia as one of the test databases for this study.

Bibliography

[BAM93] M. Berry, S. Allen, and R. MacIntyre. XLSI: A motif-based user interface for a conceptual retrieval system. *The X Journal*, 3(2):58–64, 1993.

[Ber92] M. W. Berry. Large scale singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[DDF+90] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[Dum91] S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, 23(2):229–236, 1991.

[GL89] G. Golub and C. Van Loan. *Matrix Computations*. Johns-Hopkins, Baltimore, second edition, 1989.

[SB90] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[Shi95] A. T. Shippy. An Analysis of the Latent Semantic Indexing Retrieval System. Master's thesis, The University of Knoxville, Tennessee, Knoxville, TN, May 1995.

[WS90] L. Wall and R.L. Schwartz. *Programming Perl*. O'Reilly & Associates, Inc., Sebastopol, CA, 1990.

Student Observation Checklist

The students were to include the answers for the following questions in their weekly reports. The entire set of questions were answered once for each patron observed. The students were also instructed to answer this same question set for queries that they performed themselves. The questions are worded exactly as they were given to the students.

1. What were the queries? (Answer remaining questions per query!)
2. How were the queries formulated (mouse, typing, relevance feedback, hybrid)?
3. How successful were the queries? (Can you tell if they got the information they wanted?)
4. What errors with LSI were detected? (Missed documents or keywords?)
5. What errors with the interface were detected?
6. How often did they request help? What type of help?
7. What problems did they have in searching? (Couldn't think of search terms, terms, poor results, couldn't understand relevance feedback or document-based searching, etc.)
8. Why were they using the system? (Playing? Trivial Pursuit? Research?)
9. What database (CCE or KNOXNS) did they use?
10. Did they have any comments or suggestions for improvements?
11. Did they attempt to print any text? How much? Any problems?

In addition to the eleven questions above, the students were required to provide a summary of the observations, give constructive comments on the system, and report any other information that they felt to be pertinent.

Student Homework Assignments

Examples of queries from the weekly student homework assignments appear below. Since the type of queries in the assignments varied, sample queries for each week are provided. The database searched is denoted with the abbreviation *CCE*, for the Concise Columbia Encyclopedia, or *KNOXNS*, for the Knoxville News Sentinel. The queries are worded exactly as they were presented to the students.

Week 1 (CCE)

- What is the name for the religious and ethical duties of the individual in Hinduism?
- What was the 1972 court case in which the U.S. Supreme Court ruled that Amish children could be exempted from compulsory school attendance beyond the 8th grade?
- What music term was originally used to indicate that a certain part was indispensable to the music?
- What type of lens, which is thicker at the edges than at the center, bends parallel light rays passing through it away from each other?

Week 2 (CCE)

- What U.S. Senator was expelled for helping the British seize Spanish Florida?
- Father of the actor who starred in Gunga Din (1939)?
- What are the larvae of harvest mites called?
- What is the largest reservoir in the U.S.?

Week 3 (CCE)

- What American admiral uttered the famous cry "Damn the torpedoes mines!"?
- Who was the first vice president to succeed in presidency?
- In what game is the object to punch, dribble, or kick the ball into or directly over the goal?
- What term describes the position of three planets along a straight line?

Week 4 (KNOXNS)

- "Read my lips, no new taxes."
- "Mother of all battles."
- "A thousand point of light."
- "Kinder gentler nation."

Week 5 (KNOXNS)

- What was the average age of a Desert Storm soldier? Average age of a Vietnam soldier?
- Who said "We have grasped the mystery of the atom and rejected the sermon on the mount"?
- What is the date of the Iraqi invasion of Kuwait?
- Who said "It is war's prize to take all vantage"?

Week 6 (KNOXNS)

- How much hazardous waste did TN ship across the state line in 1989?
- What former UT student gave a workshop on hazardous materials at the UT Conference Center during the month of March in 1991?
- What type of dogwood tree has multi-colored leaves with red blossoms?
- What is the name of a Blount County recycling program that raised over $1000.00 for a girls' home in 1991?

Week 7 (CCE and KNOXNS)

- On average, what percent of the human body is water?
- What is the most important step in treating our water supplies before we use them?
- How many gallons of water will one gallon of gasoline contaminate?
- How many gallons of water and liquids are recycled through your kidneys each day?

Week 8 (CCE and KNOXNS)

- How much oil does the United States import from Kuwait each day?
- What makes an airplane fly?
- What characteristic determines whether a tree is deciduous or a conifer?
- What is the University of Tennessee football team's record at the Sugar Bowl?