DYNAMIC CATEGORIZATION: A METHOD FOR DECREASING INFORMATION OVERLOAD

A DISSERTATION
SUBMITTED TO THE PROGRAM IN
MEDICAL INFORMATION SCIENCES
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Wanda M. Pratt
March 1999

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Russ B. Altman, Principal Adviser
(Departments of Medicine and Computer Science)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Lawrence Fagan
(Department of Medicine)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Marti Hearst
(School of Information Management & Systems,
University of California, Berkeley)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Edward H. Shortliffe
(Departments of Medicine and Computer Science)

Approved for the University Committee on Graduate Studies:

Abstract

Search results can be overwhelming. When people use computer-based tools to find answers to general questions, they often are faced with a daunting list of search results or “hits” returned by the search engine. Many search tools address this problem by helping users to make their searches more specific. However, when dozens or hundreds of documents are relevant to their question, users need tools that help them to explore and to understand their search results, rather than ones that eliminate a portion of those results.

I have developed a new approach, called dynamic categorization, that addresses this problem by automatically organizing search results into meaningful groups that correspond to the user's query. This approach uses knowledge of important kinds of queries and a model of the domain terminology to generate a hierarchical categorization of search results. I created a tool called DynaCat that implements this approach for the domain of medicine, where the amount of information in the primary medical literature alone is overwhelming. DynaCat summarizes the documents returned from a search by organizing them into an intuitive and useful hierarchy of categories, thus helping patients as well as health-care workers to gain quick and easy access to important medical information.

I evaluated my thesis work in two ways. The technical evaluation demonstrated that the categorization generated by DynaCat was about as consistent with the physicians' categorizations as the physicians' categorizations were with each other. These results suggest that DynaCat creates reasonable document categories and assigns documents to categories appropriately. In the usefulness evaluation, I showed that breast cancer patients and their

family members could find more answers in a fixed amount of time, and were more satisfied with their search experience when they used DynaCat than when they used either the cluster tool or the ranking tool. These differences were statistically significant ($p < 0.05$). Users thought that DynaCat helped them to find answers easily and quickly, and to learn about the information related to their query. They indicated that DynaCat provided an organization of search results that was clear, easy to use, accurate, precise, and helpful.

Acknowledgements

In 1988, shortly after I finished my bachelor's degree, I read the book on MYCIN by Ted Shortliffe and Bruce Buchanan. I was hooked; I had found a field that I knew would both excite me and challenge me for many years. I dreamed of going to Stanford's Medical Informatics Program, and the reality has turned out to be as good as I had imagined.

Many people have made my years at Stanford rewarding. I thank Ted Shortliffe for creating an outstanding medical informatics program, for providing me with solid research advice, and for promoting my career. I am grateful to Larry Fagan for being a kind, thoughtful, and thorough advisor. Through his guidance, I learned how to refine my research ideas, and how to communicate those ideas clearly to others. I thank Russ Altman for being a supportive research and academic advisor. His enthusiasm, insightful comments, and sympathetic ear helped me through the rough spots in my graduate training. I thank Marti Hearst for her meticulous scrutiny of my dissertation. I am grateful to her for our many discussions, especially when I was formulating my thesis ideas. I thank Lyn Dupré for editing this dissertation. I am even more grateful to her for teaching me to write well, and for helping me to find pleasure in what used to be a miserable task. I thank Mike Walker for helping me to design my evaluations, to analyze the results, and to learn how to do everything on my own next time.

I feel very fortunate to have been a student in Stanford's Medical Informatics program. The excellent faculty, students, and staff have created a environment that promotes absorbingly interesting discussions, exciting research projects, and supportive friendships. I am

grateful to Darlene Vian for being a great listener, travel consultant, and party provider. I thank Gretchen Purcell, Smadar Shiffman, Maria Tovar, Gillian Sanders, Ida Sim, Doug Fridsma, Diane Oliver and the many other fantastic students who made my years at Stanford intellectually stimulating and fun. I am especially grateful to my classmate Ramon Felciano for the lively philosophical debates, exhilarating brainstorming sessions, sympathetic conversations, and wild parties that helped me enjoy and survive my graduate training.

Thanks to everyone at Lexical Technology, Inc. for allowing me to use their search engine and their tools for accessing the UMLS. I am particularly grateful to Kevin Keck for adding the functionality that I needed and for quickly responding to my numerous requests. I appreciate the support of everyone on the Agents project. I thank Gio Weiderhold for providing insightful feedback on my presentations and for being an enthusiastic supporter of my research.

I extend my gratitude to the Stanford Health Library, the Community Breast Health Project, and Bob Carlson for helping me to recruit breast-cancer patients and their family members for one part of my evaluation, and I thank Bob Carlson for helping me recruit physicians for the other component of my evaluation. I am grateful to the many subjects who participated in my studies, particularly Patti Schank, Sally Mouzon, and Lanette Hendren, who donated their time to participate in the pilot study and who provided me with valuable feedback. I thank Lauren Langford at the Community Breast Health Project for gathering the list of frequently-asked questions about breast cancer. Many thanks to Mehran Sahami for allowing me to use his system as the cluster tool in my evaluations.

My research was supported in part by grant LM-07033 from the National Library of Medicine and by contract N44-CO-61025 from the National Cancer Institute.

Thanks to Mike Pazzani and the rest of my colleagues at UC-Irvine for motivating me to finish my Ph.D. quickly by providing me with an exciting job, and for understanding when it took me longer to finish than I expected.

I am grateful to my friends who kept me sane during my graduate training. I thank Patti Schank for being an excellent, empathetic friend, whenever I needed her. I thank my many other friends: Larry Hamel, Doug Fridsma, Lisa Fridsma, Kevin Thompson, Sally Mouzon, Rich Acuff, and Amy Guthrie for remaining my friends, even when I rarely had time to be with them. I am also grateful to my cat Spook for keeping my blood pressure down by insisting that I take regular breaks from writing to snuggle him.

I am fortunate to have been raised in a loving and supportive family. My grandmother, Nellie Pratt, and I have always had a special connection, and I thank her for being there for me. I thank my aunt, Pat Niere, for demonstrating that it is never too late to pursue your dreams. I am grateful to my cousin, Richard Hallett, for going through the Ph.D. process first and helping me through a few of the rough spots. I am most indebted to my parents. I am grateful to my mom, Emolyn Rollin, for raising me in a warm and supportive environment and for making my early education a top priority. I thank my dad, Victor Pratt, for teaching me to think logically and for always believing that I am capable of doing anything that I set my mind to.

Most of all, I want to thank my husband, John Gennari. As my colleague, he helped me find the right words and pictures to communicate my thoughts clearly. As my partner and best friend, he knew when to encourage me, when to challenge me, and when to comfort me. Over the past many months, he has done more than his fair share of everything (from wedding planning, to packing and moving) without a complaint. I cannot imagine surviving the Ph.D. process without his support. I dedicate this dissertation to him.

Table of Contents

Abstract

Acknowledgements

Table of Contents

List of Tables

List of Figures

Chapter 1 — Organization of Search Results

1.1 Search Process 2

1.2 Search Scenario 4

1.3 Support for Understanding and Exploring Search Results 6

1.4 Desirable Characteristics for Organizing Documents 7

1.4.1 Assignment of Meaningful Labels 8
1.4.2 Document Groups Responsive to Search Results 8
1.4.3 Query-Sensitive Document Groups 8
1.4.4 Placement of Documents in All Appropriate Groups 9

1.5 Characteristics of Previous Approaches to Organizing Documents 9

1.6 Research Hypothesis 11

1.6.1 Technical Claim 12
1.6.2 Usefulness Claim 12

1.7 Dynamic Categorization: An Approach to Organizing Search Results . . . 12

1.7.1 Domain Models 13
1.7.1.1 Terminology Model 15
1.7.1.2 Query Model 15
1.7.2 Categorizer 16
1.7.3 Organizer 17
1.7.4 Interfaces 18
1.7.5 Example Use of DynaCat 18

1.8 Evaluation 20

1.8.1 Evaluation of Usefulness Claim 20
1.8.2 Evaluation of Technical Claim 22

1.9 Guide for the Reader 23

Chapter 2 — Previous Approaches to Organizing Documents

2.1 Document Representations 26

2.1.1 Vector Space 26
2.1.2 Controlled Vocabulary 28
2.1.3 Structured Documents 29
2.1.3.1 Document Components 29
2.1.3.2 Structured Abstracts 30
2.1.3.3 Context Models 30

2.2 Query Representation 31

2.2.1 Boolean Queries 31
2.2.2 Vector-Space Queries 32
2.2.3 Natural-Language Queries 32
2.2.4 Documents as Queries 32

2.3 Relevance Ranking 33

2.3.1 Similarity Scoring 33
2.3.2 Use in Presentation of Search Results 35
2.3.3 Advantages and Disadvantages of Relevance Ranking 36

2.4 Document Clustering 39

2.4.1 Document-Clustering Algorithms 40
2.4.2 Feature Selection 41
2.4.3 Use in Matching Documents to Queries 42
2.4.4 Use in Presentation of Search Results 42
2.4.5 Advantages and Disadvantages of Document Clustering 44

2.5 Document Classification 45

2.5.1 Automated Algorithms 45
2.5.1.1 Feature Selection for Supervised-Learning Techniques 46
2.5.2 Use in Presentation of Search Results 46
2.5.3 Advantages and Disadvantages of Document Classification 47

2.6 Summary and Comparison to Dynamic Categorization 47

Chapter 3 — System Specification

3.1 Query Model 50

3.1.1 Query Types 51
3.1.2 Category Types 52
3.1.3 Creation of the Query Model 53

3.2 Terminology Model 55

3.2.1 Medical Terminology Model 55
3.2.2 Terminology Model Requirements 57
3.2.2.1 Terminology Models for Other Domains 58

3.3 Categorizer 59

3.3.1 Current Approach: Keyword Pruning 59
3.3.1.1 Implications for Terminology-Model Requirements 59
3.3.1.2 The Keyword-Pruning Algorithm 61
3.3.1.3 Example of the Categorization Process 61
3.3.2 tExploratory Approaches 62
3.3.2.1 Title-Term Spotting 63
3.3.3 Exploratory Approaches 65
3.3.3.1 Title-Term Spotting 65
3.3.3.2 Information Extraction 68

3.4 Organizer 71

3.4.1 Additional Requirements for the Domain Models 71
3.4.2 Organizer Algorithm 72

3.5 Results-Presentation Interface 76

3.6 Summary 77

Chapter 4 — Usefulness Evaluation

4.1 Objectives 80

4.2 Comparison Systems 80

4.2.1 Relevance-Ranking Tool 82

4.2.2 Clustering Tool 83
4.3 Pilot Study 84
4.4 Final Study 84
4.4.1 Methods 84
4.4.1.1 Subjects 85
4.4.1.2 Procedure 85
4.4.2 Results 89
4.4.2.1 Timed Tasks 89
4.4.2.2 Amount Learned 91
4.4.2.3 User Satisfaction 92
4.4.2.4 Comments and Answers to Open-Ended Questions 98
4.5 Summary 99

Chapter 5 — Evaluation of Technical Claim

5.1 Objectives 101
5.2 Pilot Study 102
5.3 Final Study 104
5.3.1 Methods 104
5.3.1.1 Data Sets 104
5.3.1.2 Subjects 105
5.3.1.3 Procedure 105
5.3.1.4 Metrics 106
5.3.2 Results 110
5.3.2.1 Consistency 111
5.3.2.2 Accuracy 114
5.3.2.3 Subjective Assessment 116
5.4 Summary 116

Chapter 6 — Summary and Conclusions

6.1 A Knowledge-Based Approach to Organizing Search Results 119
6.2 Contributions 120
6.2.1 Information Access 121
6.2.2 Knowledge-Based Systems 121
6.2.3 Medicine 122
6.3 Limitations 122

6.3.1 Scope of Query Model 122
6.3.2 Domain-Modeling Effort 123

6.4 Future Work 123

6.4.1 Interactive Categorization Environment 124
6.4.2 Information Filtering 124
6.4.3 Categorization of Informal Medical Information 124

6.5 Concluding Remarks 125

Appendix A — User-Satisfaction Questionnaire 127

Appendix B — Frequently Asked Questions About Breast Cancer 131

Appendix C — Tutorial for Category Tool 141

Appendix D — Tutorial for Cluster Tool 145

Appendix E — Tutorial for Ranking Tool 149

Appendix F — Timed Tasks for Each Query 151

Appendix G — Instructions for Organizing Documents 153

Appendix H — View of Categories in Evaluation of Technical Claim 157

Appendix I — View of Clusters in Evaluation of Technical Claim 159

Bibliography

List of Tables

1.1 Comparison of desirable characteristics of three automated approaches to organizing documents 11

3.1 Patient-oriented medical query types and their typical forms 52

3.1 Categorization criteria for the query type treatment—problems 64

3.2 Example keywords corresponding to the citation shown in Figure 3.4. The keyword in bold would become the category label for this citation 64

3.3 Example of how the terms in the title “Angiosarcoma of the Breast Following Segmental Mastectomy Complicated by Lymphedema” could be used to categorize that document for a query on the complications of a mastectomy. 67

4.1 Results for the timed tasks 90

4.2 Subjects’ comments on DynaCat 98

4.3 Subjects’ comments on the cluster tool 99

4.4 Subjects’ comments on the ranking tool 99

5.1 Interpretation of the kappa statistic 108

5.2 Contingency table for the assignment of documents to a category 109

5.3 Summary of consistency results for the cluster tool 113

5.4 Comparison of the maximum and average number of documents assigned to a category by the study subjects versus by DynaCat 114

5.5 Average scores for the subjects' assessment of the desirable characteristics for categories and clusters 116

List of Figures

1.1 The search process 3

1.2 PubMed interface corresponding to a search on the prevention of breast cancer . . 4

1.3 DynaCat's interface 5

1.4 The search process when dynamic categorization has been incorporated 14

1.5 DynaCat's initial display for a query on the adverse effects of a mastectomy . . . 19

1.6 DynaCat's display after scrolling down the left window pane 19

1.7 DynaCat's display after clicking on the hyperlink corresponding the Hemic And Lymphatic Diseases category 20

2.1 Interface that allows the user to adjust relevance-ranking criteria for the Lycos Pro web-based search engine (Lycos 1997) 36

2.2 Lycos relevance-ranking interface 37

2.3 TileBars interface 38

2.4 Scatter/Gather interface 43

3.1 DynaCat's system architecture 50

3.2 The connection between the terminology model and the query model 56

3.3 Flow diagram for keyword pruning 62

3.4 Example citation returned from a search on *mastectomy* and *adverse effects* *63*

3.5 Example citation returned from a search on *mastectomy* and *adverse effects* *63*

3.6 Flow Diagram for title term spotting 66

3.7 An example of concept node for the query type *disease—prognostic indicators* . *70*

3.8 Example titles from search on skin cancer prognosis 70

3.9 Example of the MeSH ancestry tree for the original category labels (indicated by the document icon). 75

3.10 Organization for a maximum breadth threshold of four 75

3.11 Organization for a maximum-breadth threshold of three. 76

3.12 DynaCat's interface 77

4.1 The interfaces to DynaCat (a), the cluster tool (b), and the ranking tool (c) 81

4.2 The usefulness evaluation study design. 86

4.3 Results from the first five questions from the validated user-satisfaction questionnaire 92

4.4 Results from the second five questions from the validated user-satisfaction questionnaire 93

4.5 Results for my user-satisfaction questionnaire 94

4.6 Results of yes—no user-satisfaction questions 95

4.7 Results for questions regarding which tools helped the subjects to learn 97

4.8 Results for the question regarding the tool that subjects liked least. 97

4.9 Results for the question regarding the tool that subjects liked best. 98

5.1 DynaCat's interface for a search on the complications of a mastectomy 102

5.2 Pseudo code for calculating *Pchance*. *108*

5.3 Intercategorizer and DynaCat-categorizer consistency for the prevention-of-breast-cancer search results. 111

5.4 Intercategorizer and DynaCat-categorizer consistency corresponding to the query

on the prognostic indicators for breast cancer 112

5.5 Intercategorizer and DynaCat-categorizer consistency corresponding to the query on diagnostic tests for breast cancer 112

5.6 Average precision and recall in comparisons of DynaCat to the test subjects, and the subjects to each other 115

5.7 Average precision and recall in comparisons of the cluster tool to the test subjects, and the subjects to each other 115

Chapter 1

Organization of Search Results

The amount of information available to the general public has been growing rapidly for many decades. During the 1800s, the number of scientific publications doubled every 50 years; in this century, it has doubled every 10 to 15 years (Warren 1981). In the past few years, the World Wide Web has facilitated an explosion of informal information as well.

As the volume of both formal and informal information available increases, people become overwhelmed by the amount of information. They become frustrated when their searches yield tens or hundreds of relevant documents, so they abandon their search before they understand the kinds of information that it has returned. In my thesis work, I propose that a solution to this problem is to organize the documents returned from a search into meaningful groups that correspond to the query. I have developed a new approach that automatically generates such an organization of documents. I implemented this approach for the domain of medicine, where the amount of information in the primary medical literature alone is overwhelming. For example, **MEDLINE**, an on-line repository of medical abstracts, contains more than 9.2 million bibliographic entries from over 3800 biomedical journals; it adds 31,000 new entries each month (NLM 1998a).

In this dissertation, I describe how my approach provides information about (1) what kinds of information are represented in (or are absent from) the search results, by creating document categories with meaningful labels and by hierarchically organizing the document categories; (2) how the documents relate to the query, by making the categorization dependent on the type of query; and (3) how the documents relate to one another, by grouping ones that cover the same topic into the same category. This approach summarizes the documents returned from a search into an intuitive and useful hierarchy of categories, thus helping patients as well as health-care workers to gain quick and easy access to important medical information.

1.1 Search Process

When people use a computer to search for information, they normally follow three basic steps:

1. Formulating the query
2. Receiving documents that match the query
3. Understanding the search results

First, they express their information need as a **query**, a representation that the search engine can use to find matching documents. This step in the search process is often called **query formulation**. The form of the query can vary from system to system; a query could be expressed in natural language, as Boolean expression of words, as a list of words, or even as an example document.

In the second step, the search engine finds documents that match its representation of the user's query. In the third step, those **search results** (the document summaries returned from the search engine) are presented to the user. The user examines the presentation and tries to use it to gain a high-level understanding of the search results. She wants to determine which documents are relevant, and to what extent those documents meet her information need. The user may repeat the search process if her information need evolves as

she learns more about what information is available or how she can express her information need using the system. This search process is illustrated in Figure 1.1.

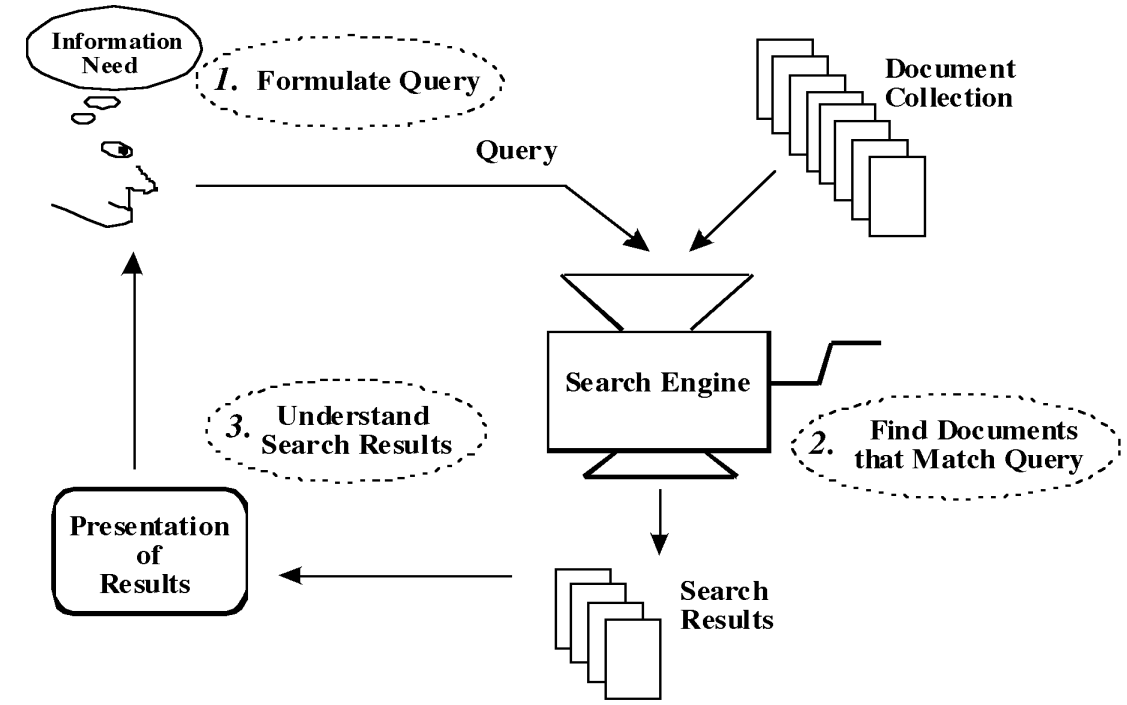


Figure 1.1 — The search process. A user must try to express her information need as a query that the search engine can process. The search engine returns documents that match the user's query. It presents some representation of those documents to the user. The user must use that presentation of results to determine what information is present in the search results and how that information meets her need. If she does not find what she wants or if her conception of the information need changes, she can reformulate the query for a new search.

Much of the research in information retrieval and information access has addressed the first two steps in the search process: (1) formulating and reformulating queries, and (2) matching documents to the query. Relatively little research has been done on the third step: presenting the search results in a way that helps users to understand and explore their search results. This third step is the focus of my research.

1.2 Search Scenario

Consider a woman whose mother has been diagnosed recently with breast cancer. She is worried about her own chances of developing breast cancer, and she wants to know what she can do to prevent breast cancer. She has read a few options in patient information pamphlets, but she wants to see more detailed and recent information, such as that in medical journal articles.

She could choose to search the primary medical literature using PubMed (NLM 1997c)—the free, web-based MEDLINE search tool. If she searches for documents in the previous year that use the keywords *breast neoplasms* and *prevention* anywhere in the document, PubMed returns the titles of over 400 documents displayed as a long list (see Figure 1.2).

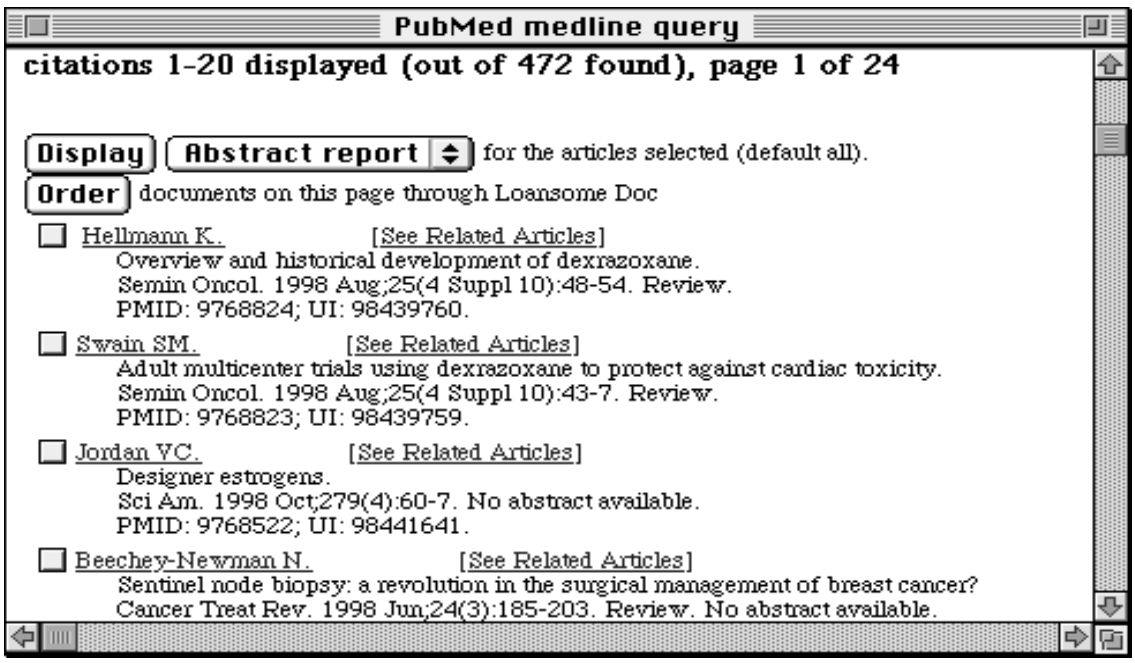


Figure 1.2 — PubMed interface corresponding to a search on the prevention of breast cancer. The search results are displayed in a long list, sorted chronologically. The user may choose which type of information is shown about each document (citation report, abstract report, and so on), but there is no way for her to group the documents or even to change the ordering of the documents in the list.

If the user notices a document title that she finds interesting, she can find related documents using the *See Related Articles* link, but she cannot see a summary of the information contained in those search results. If she wants to form an accurate model of all possible preventive measures, she must examine all 472 documents. Even if she spends only 30 seconds examining each document, it will take her nearly 4 hours to browse the entire list of search results.

In contrast, if she were to use **DynaCat**, the document-categorization tool that I developed, she could see the search results organized by the preventive actions found in those documents. Figure 1.3 shows the interface generated by DynaCat for a search on the CancerLit database (NLM 1997b) using the keywords *breast neoplasms* and *prevention*[1].

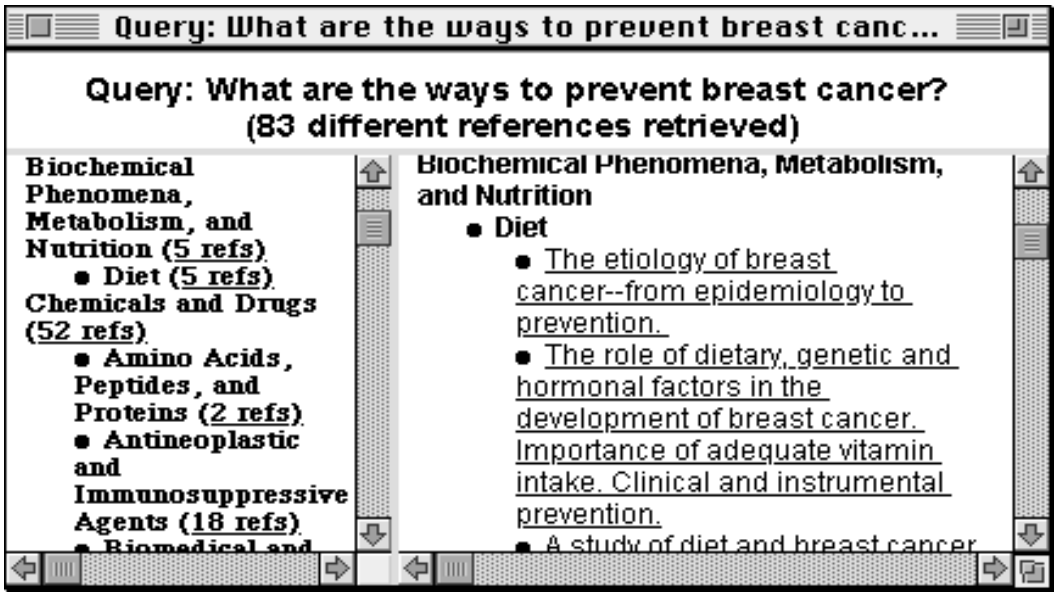


Figure 1.3 — DynaCat's interface. The interface is divided into three frames, or window panes. The top window pane displays the user's query and the number of documents found. The left pane shows the categories in the first two levels of the hierarchy. This pane provides a table-of-contents view of the organization of search results. The right pane displays all the categories in the hierarchy and the titles of the documents that belong in those categories.

1. DynaCat used a search engine that accessed only the CancerLit database, which is a subset of the MEDLINE database. Fewer documents were found in the CancerLit database than were found in the MEDLINE database.

By organizing the documents into a hierarchy of categories that represent the preventive actions discussed, this interface helps the user to learn about the various preventive measures that are discussed in the literature. For example, she can determine immediately that five documents discuss diet as a preventive measure. This organization of results also helps her to find information about specific preventive measures quickly and easily.

1.3 Support for Understanding and Exploring Search Results

Regardless of how the user formulates her query or how the search engine matches documents to queries, the user needs to understand her search results. The point of the search process is to find information that satisfies the user's information need, but the user also may need help in assessing whether or how the retrieved documents meet her needs.

Current information-retrieval tools usually return a simple list of documents as the search results. However, people can become overwhelmed by and have difficulty assessing the kinds of information available in such lists.

Most search tools assist in solving this problem by helping a user to formulate a more specific query. However, even if she could express her information need perfectly to the search engine, and even if the search engine found only documents that were relevant to the query, the user might still need tools to help her to understand what kinds of documents have been returned. By focusing on query formulation, search-tool developers seem to assume that the documents that are relevant to the user's information need will be few; however, many documents may be truly relevant. The user may have a broad information need, or the document collection being searched may contain many documents covering the user's information need. If the user specifies a more specific query, she may eliminate relevant documents.

As the user examines documents related to her query, her information need may change as she learns more about the information that is available. The user may be able to reformulate a query to match her new information need; however, if she does not understand the extent of the information available in her search results, her ability to reformulate the query may be impeded. For example, a patient with breast cancer may become interested in arthritis when she discovers that that disorder is a possible complication of a mastectomy, but if the document discussing arthritis is buried in a long list of other documents, the user may never notice that there is such a complication. As psychological studies have shown (Crowder 1976), it is much easier for a person to recognize the interest she has in information that is presented to her than it is for her to specify a priori the topics in which she is interested.

1.4 Desirable Characteristics for Organizing Documents

One way to help users to understand and to explore their search results is to organize those results into groups of documents. Documents can be grouped in multiple ways, some of which are more useful than others for helping users to understand their search results. Ideally, we would like empirical evidence on the characteristics of document groupings that are most important for this task. Unfortunately, few researchers have pursued such investigations. However, a few characteristics seem intuitively desirable. A document-grouping tool should:

1. Assign meaningful labels to the document groups.
2. Create document groups that are responsive to the content of the documents in the search results.
3. Create document groups that correspond to the user's query.
4. Place documents in all appropriate groups.

I describe each of these characteristics in turn in Sections 1.4.1 through 1.4.4.

1.4.1 Assignment of Meaningful Labels

If the user is to understand the information that is represented in the search results, the documents must be organized into groups that have **meaningful labels**. A label is meaningful if it describes succinctly a common theme among the documents in a way that allows most people from the target user group to understand the definition of that label and to determine which documents belong in a group with that label. If the document groups have meaningful labels, the user can assess quickly the contents of each document group and can thus determine in which groups she is likely to be interested.

1.4.2 Document Groups Responsive to Search Results

The categorization process should be data driven. That is, the labels should not be predefined, but rather should be generated by characteristics of the documents in the search results. The categorization should provide a topic summary of the documents contained in the search results, rather than listing prespecified document groups. If the groups are prespecified, the person specifying the groups may forget to include rare possibilities, or new possibilities may arise after the possible groups have been specified. In medicine, our knowledge about diseases, treatments, complications, preventive measures, and so on changes rapidly, making it difficult for someone to maintain an up-to-date list of possible document groups for any possible query.

1.4.3 Query-Sensitive Document Groups

An organization of documents that is **query sensitive** uses only document groups that correspond to the user's query, such that the labels could be considered complete or partial answers to the query. This characteristic is different from that of assigning meaningful labels to all possible documents groups: Such labels may be meaningful, but may not address the user's query. There are many ways to group documents that would be meaningful but would not be useful or interesting to the user. For example, we could group the search results for breast cancer prevention into the documents that discuss characteristics

of a study (e.g., type or duration), characteristics of the subjects (e.g., pre-menopausal women or women with a family history of breast cancer), or the type of article (e.g., review, clinical trial, or editorial) but none of those topics directly address the user's query. If the documents are grouped in all possible ways, the user may find the number of groupings more overwhelming than the raw search results. A useful organizational tool creates a categorization structure that corresponds to the user's query.

1.4.4 Placement of Documents in All Appropriate Groups

When a document discusses more than one topic that relates to the user's query, a categorization system should associate the document with all the appropriate topics, rather than determining only one primary topic. Most clustering and classification systems enforce a partition on the document groups, and, thus, do not place any document in more than one category. However, many documents discuss more than one topic. For example, a document entitled "The role of Tamoxifen and diet in the prevention of breast cancer" is likely to discuss two, different preventive measures for breast cancer: dietary modification and Tamoxifen. Because the user explores the search results by selecting document groups of interest to her, any document-grouping tool should place all documents that discuss a topic in that topic's corresponding group. To avoid misleading the user, tools also should place documents in only those groups that correspond to topics that are present in the document.

1.5 Characteristics of Previous Approaches to Organizing Documents

Automatic approaches to organizing documents include relevance ranking, clustering, and classification. These techniques typically represent each document as a vector of the words that appear in the document.

Relevance-ranking systems create an ordered list of search results. The order of the documents is based on a measure of similarity between the document and the query; that similarity measure is used as an approximation of the relevance of the document to the query

(van Rijsbergen 1979; Salton 1989; Harman 1992). Yet, an ordered list does not give the user information about the similarities or differences in the content of the documents. For example, the user would not be able to determine that 30 different preventive measures were discussed in the retrieved documents, or that 10 documents discussed the same method. People usually do not have the time to browse all the documents on the list. They may give up examining the documents long before they see all the results, and thus may miss useful information.

Document-clustering systems create groups of documents based on associations among the documents (Willett 1988; Rasmussen 1992; Hearst and Pedersen 1996; Sahami 1998). To determine the degree of association among documents, clustering systems require a **similarity metric**, such as the number of words that the documents have in common. The systems then label each group (or **cluster**) with that group's commonly occurring word or words. Unfortunately, the similarities found by clustering may not correspond to a grouping that is meaningful to the user. Even if the grouping is meaningful to the user, it may not correspond well to the user's query because clustering algorithms do not use any information about the user's query in forming the clusters. Also, since the document groups are labeled by only the frequently occurring words in the group, the user may not form a good model of the kinds of documents present in the cluster. In Section 2.4, I discuss clustering approaches in greater detail.

In contrast, **document-classification** systems use supervised-learning algorithms to create document groups; they use a training set that contains a large number of documents assigned to predefined categories (Lewis 1992a; Sahami 1998). The classifier uses the training set to infer the criteria that indicates that a document belongs to a category. Although the document groups have meaningful labels, the groups are predefined, so they cannot adapt to the user's query or to the distribution of documents in the search results. For example, if a document discusses a new preventive measure, such as taking the drug Tamoxifen, that was not one of the predefined categories in the training set, classification techniques would not be able to generate a new category for that preventive measure. I

discuss classification techniques in Section 2.5. Table 1.1 summarizes the features of these organization techniques.

Table 1.1. Comparison of desirable characteristics of three automated approaches to organizing documents.

Desirable Characteristics	Approach		
	Classification	**Clustering**	**Relevance Ranking**
Meaningful labels	yes	no	no
Document groups responsive to search results	no	yes	yes
Query-sensitive categorization	no	no	yes
Placement of documents in all appropriate groups	certain algorithms	certain algorithms	no

1.6 Research Hypothesis

I propose that the results of a broad search, in which many documents are relevant to answering the user's question, can be organized in a manner that helps the users to find answers to their query quickly and to feel satisfied with their search experience. A system can generate this organization automatically using knowledge of the user's query and a model of the domain terminology.

In support of this general hypothesis, I make two claims in my research: a technical claim and a usefulness claim. I validate these claims (1) by describing a method, **dynamic categorization**, that satisfies the criteria that I propose for the technical claim; (2) by creating a prototype system, DynaCat, that implements this method in the medical domain with patients as the targeted user group; and (3) by performing experiments to verify both claims.

1.6.1 Technical Claim

My technical claim is that dynamic categorization is a new approach to grouping documents automatically that combines the desirable characteristics of clustering with those of classification (see Section 1.4). Such an approach should:

1. Assign meaningful labels to the document groups.
2. Create document groups that are responsive to the content of the documents in the search results.
3. Create document groups that correspond to the user's query.
4. Place documents in all appropriate groups.

1.6.2 Usefulness Claim

My usefulness claim is that the application of such a system to organizing search results is more useful to users who have general questions than are two other approaches to organizing search results: relevance ranking and clustering. A useful system helps users to

1. Learn about the kinds of information that pertain to their query
2. Find answers to their question efficiently and easily
3. Feel satisfied with their search experience

Although I implemented and tested the method in only one medical domain with only patients as users, I will argue that the method could be applicable to other domains and to other user groups.

1.7 Dynamic Categorization: An Approach to Organizing Search Results

I developed an approach that automatically creates pertinent categories, assigns the appropriate documents to each category, and generates a hierarchical organization of those cate-

gories. I call this approach **dynamic categorization** because it generates both the categorization structure and the category labels dynamically. The goal of dynamic categorization is not to separate irrelevant from relevant documents, but rather to organize the user's search results such that the organization provides information about the kinds of information that are represented by the documents in those results.

The categorization generated by this approach should help users to find specific information efficiently, and to learn about the information that is available from the retrieved documents. This approach should be particularly useful when a user has a general question and is unable to use more specific search criteria, as described in the scenario in Section 1.2.

Dynamic categorization is based on three key premises:

1. An appropriate categorization depends both on the user's query and on the documents returned from the query.
2. The type of query can provide valuable information about the expected types of categories and about the criteria for assigning documents to those categories.
3. Taxonomic knowledge about terms in the document can enable useful and accurate categorization.

Dynamic categorization adds to the original search process (Figure 1.1) four components that transform the search results into a useful organization of the same documents (Figure 1.4). These components are the query model, the terminology model, the categorizer, and the organizer. I briefly describe the domain models in Section 1.7.1, the categorizer in Section 1.7.2, and the organizer in Section 1.7.3.

1.7.1 Domain Models

In the field of information retrieval, many researchers use statistical, word-based approaches. They object to knowledge-based techniques because of the time and work required to create and maintain the necessary models for each domain, yet domain-

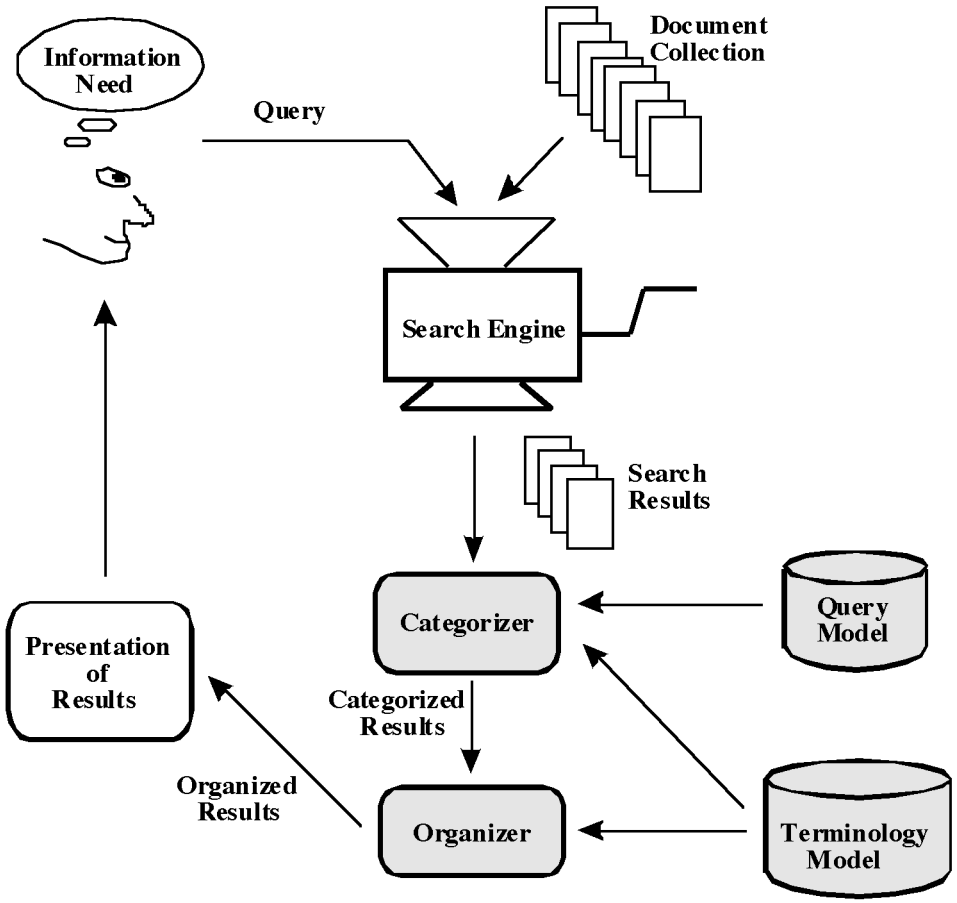


Figure 1.4 — The search process when dynamic categorization has been incorporated. The added components necessary for dynamic categorization are shown in light gray. These components do not influence which documents are returned as search results; rather, they determine how the search results are organized and displayed to the user.

specific approaches might yield superior results. Dynamic categorization is a domain-specific approach, but I have shown a case where this approach can take advantage of an existing model for much of the knowledge, rather than requiring the developer to create and maintain a large, new model. Dynamic categorization requires two types of domain knowledge. A system that implements dynamic categorization must have knowledge about the words and phrases used in those documents to organize the documents according to their medical content. This information is provided in the terminology model (see Section 1.7.1.1). The system needs knowledge about what kinds of queries users make in that domain, and about how search results from those queries should be categorized to organize the documents into categories that correspond to the user's query. The query model supplies this knowledge (see Section 1.7.1.2).

1.7.1.1 Terminology Model

The **terminology model** is a hierarchical model of domain terms, where **terms** may be single words, abbreviations, acronyms, or multi-word phrases. It is a critical component for both the categorizer and the organizer (see Sections 1.7.2 and 1.7.3). The categorizer uses the terminology model to help it infer the topics discussed in a given document. The organizer uses the terminology model to create a hierarchical organization of the categories. In Section 3.2.2, I describe the terminology-model requirements for dynamic categorization.

For the medical domain, I use the terminology model created by the National Library of Medicine, the **Unified Medical Language System (UMLS)**, which provides information on over 500,000 biomedical terms (see Section 3.2.2.1 for a list of terminology models for other domains). The UMLS links every term to at least one semantic type in a semantic network. **Semantic types** are high-level medical concepts—such as *disease or syndrome,* and *pharmacologic substance*. For example, the term *penicillin* has a semantic type of *pharmacologic substance*. The query model specifies these semantic types as part of the criteria for determining which new categories to create.

1.7.1.2 Query Model

The query model is the other critical component for the categorizer. The developer must create a query model to connect the terminology model to the categorizer, but the query model contains few concepts and requires less work to create and maintain than does the terminology model. I created the query model for DynaCat based on a list of frequently asked questions from breast-cancer patients.

To create a query-sensitive organization of the search results, dynamic categorization requires knowledge about what kinds of queries users ask, what types of categories are appropriate for those kinds of queries, and what characteristics indicate that the document belongs in a category of interest. The **query model** provides this information and the mappings that connect the information. **Query types** are high-level representations of the kinds of queries that users ask. They are independent of specific medical terms; thus, one

query type covers many specific queries. For example, both of the queries *What are the complications of a mastectomy for breast cancer?* and *What are the side effects of taking the drug Seldane to treat allergies?* have the same query type of *treatment—problems* (i.e., for a specific treatment, what problems can be encountered?), even though they specify different diseases and different treatments. The query types represent the intersection of the kinds of medical information that are available in the medical literature and the kinds of questions that users typically ask. The query model maps each query type to a **category type**, which indicates the kinds of category labels that could be assigned to the groups of documents. For example, the query type *treatment—problems* is connected to the category type *problems*, which indicates that the only appropriate category labels must be some kind of medical problem such as *infection* or *lymphedema*. For queries of the type *problem—preventive-actions*, such as *What are the ways to prevent breast cancer?*, appropriate category labels must be *preventive-actions*, such as *diet* or *Tamoxifen*.

The query model also connects each category type to **categorization criteria**, which specify the conditions that must be satisfied for a document to belong to a category of that type. I detail the query model in Section 3.1.

1.7.2 Categorizer

The **categorizer** determines what categories to create and which documents to assign to those categories. This determination can be done in different ways. Most document-classification approaches assign documents to categories based on occurrences of particular terms. For example, *lymph node* and *swollen* may be two required terms for assigning a document to the category *lymphedema*. Such term-based approaches are insufficient because terms that occur in the search results will vary tremendously with different queries of the same query type. For example, terms such as *anemia* and *ulcer* may be strong indicators of categories for a query on the adverse effects of aspirin, whereas completely different terms, such as *lymphedema* and *infection*, may be strong indicators of categories for a query on the adverse effects of a mastectomy, even though both queries are of the type *treatment—problems.* We need additional information to augment the presence or

absence of specific words. One possible source of additional information is the semantic types of the terms in the document. For example, both *anemia* and *lymphedema* have the semantic type *disease or syndrome,* even though they correspond to side effects of different treatments. Similarly, the words *infection* and *ulcer* have the same semantic type: *pathologic function*. This added knowledge about the terms in the document is one fundamental difference between previous approaches to grouping documents and dynamic categorization.

To categorize the documents, I use a technique that takes advantage of semantic-type and query-type information. I call this technique keyword pruning. **Keyword pruning** selects only those keywords that match the categorization criteria for the user's query type and creates categories with those keywords as labels. It requires that the documents have preassigned keywords, but the method through which the keywords are assigned to the documents does not matter. They could be assigned by the author of the document or by professional indexers—such as the indexers at the National Library of Medicine (**NLM**) who assign keywords, called medical subject headings (**MeSH**), to MEDLINE documents.

1.7.3 Organizer

The goal of the category organizer is to create a hierarchical organization of the categories that is neither too broad nor too deep, as defined by preset thresholds. The organizer produces the final categorization hierarchy based on the distribution of documents from the search results. When the number of categories at one level in the hierarchy exceed a preset threshold, the categories are grouped under a more general label. DynaCat generates the more general label by traversing up the MeSH term hierarchy to find a term that is a parent to several document categories.

1.7.4 Interfaces

I developed the results-presentation web-based interface for DynaCat using the Common LISP hypertext transfer protocol (**CL-HTTP**) available from the Massachusetts Institute of Technology (**MIT**) (MIT 1997). The results-presentation interface generates a web document from the categorization hierarchy produced by the organizer (see Figure 1.3).

1.7.5 Example Use of DynaCat

Consider a woman who has breast cancer. She is contemplating having a mastectomy and is worried about possible complications. She issues the query: *What are the possible adverse effects of a mastectomy?* to DynaCat and specifies her query type as *treatment—problems*. One of the categorization criteria for that query type stipulates that the keywords must be a *disease or syndrome*. If DynaCat finds a document with the keywords *lymphedema*, *arthritis*, *diagnostic imaging*, and *middle age*, the system categorizes that document under both *lymphedema* and *arthritis* because they are diseases that match the categorization criteria. It does not categorize it under *diagnostic imaging* or *middle age*, because those terms are not *diseases or syndromes*, and thus do not match the categorization criteria. Note that *lymphedema* and *arthritis* were not predefined category labels in the query model; rather, they were generated dynamically because they satisfied the categorization criteria in the query model. DynaCat's output for this user's search appears in Figure 1.5. She can immediately determine that 5 documents discuss adverse effects that are *Bacterial and Fungal Diseases*, or that 3 documents discuss adverse effects that are *Cardiovascular Diseases*. She can scroll down the left window pane and examine other high-level types of adverse effects that were found in the search results (see Figure 1.6). If she wants to know more about a particular category such as *Hemic and Lymphatic Diseases*, she can click on the adjacent hyperlink, which brings that section of the categorization hierarchy to the top of the right window pane (see Figure 1.7). If she sees an interesting document, she can click on that document's title, and its citation (including an abstract of the article) will appear in the right window pane.

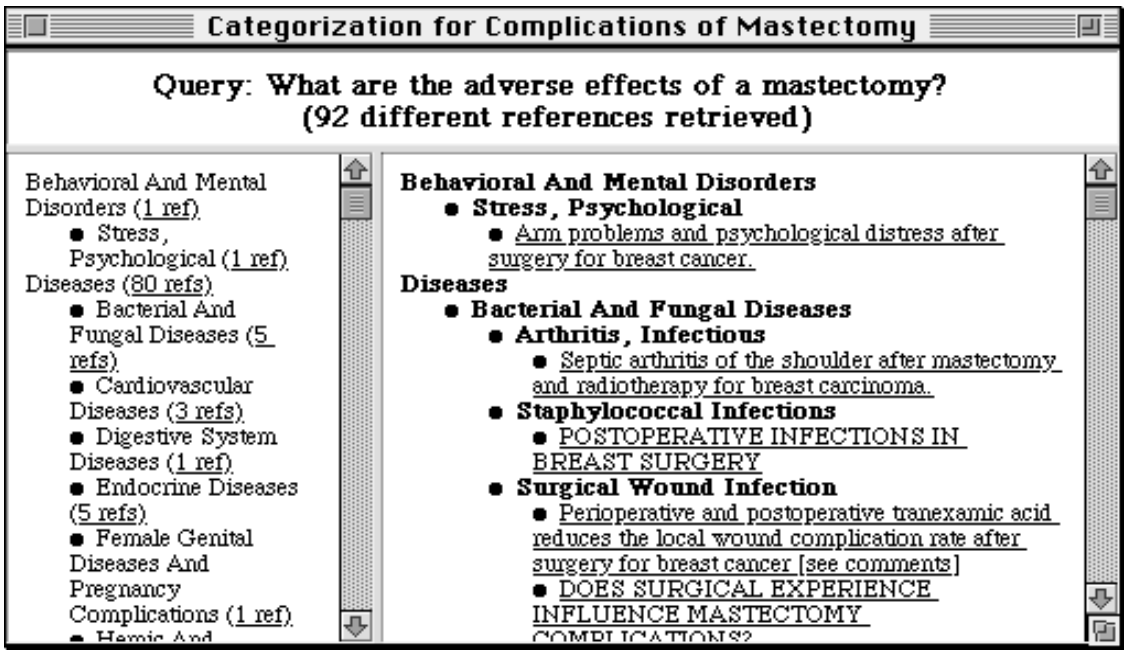


Figure 1.5 — DynaCat's initial display for a query on the adverse effects of a mastectomy.

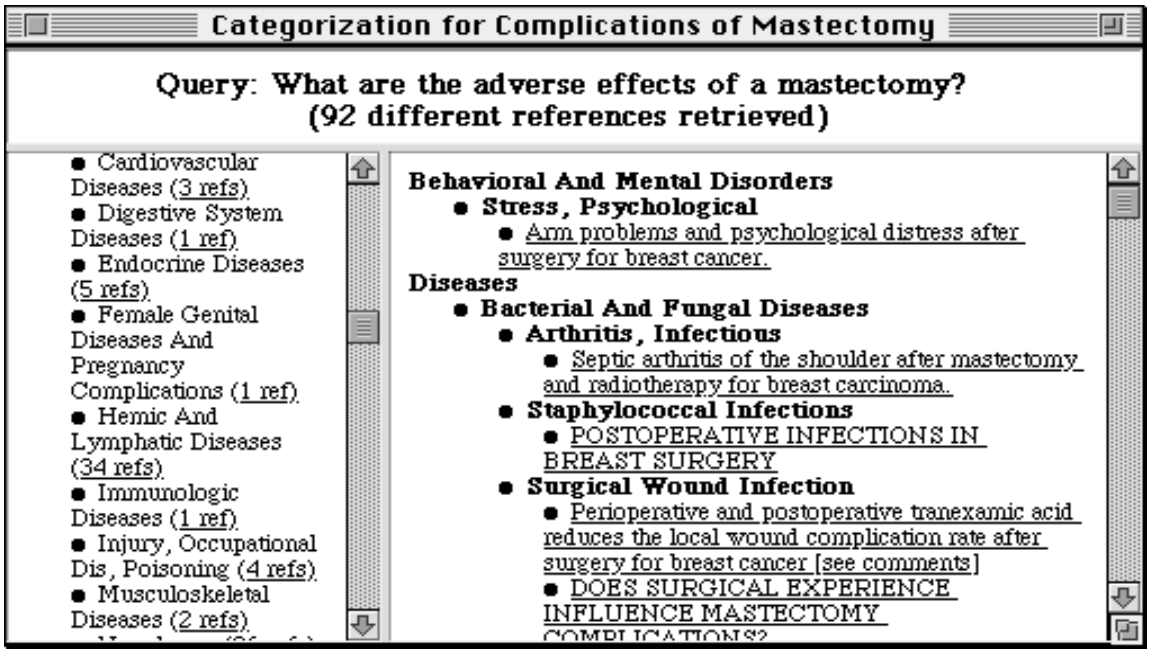


Figure 1.6 — DynaCat's display after scrolling down the left window pane.

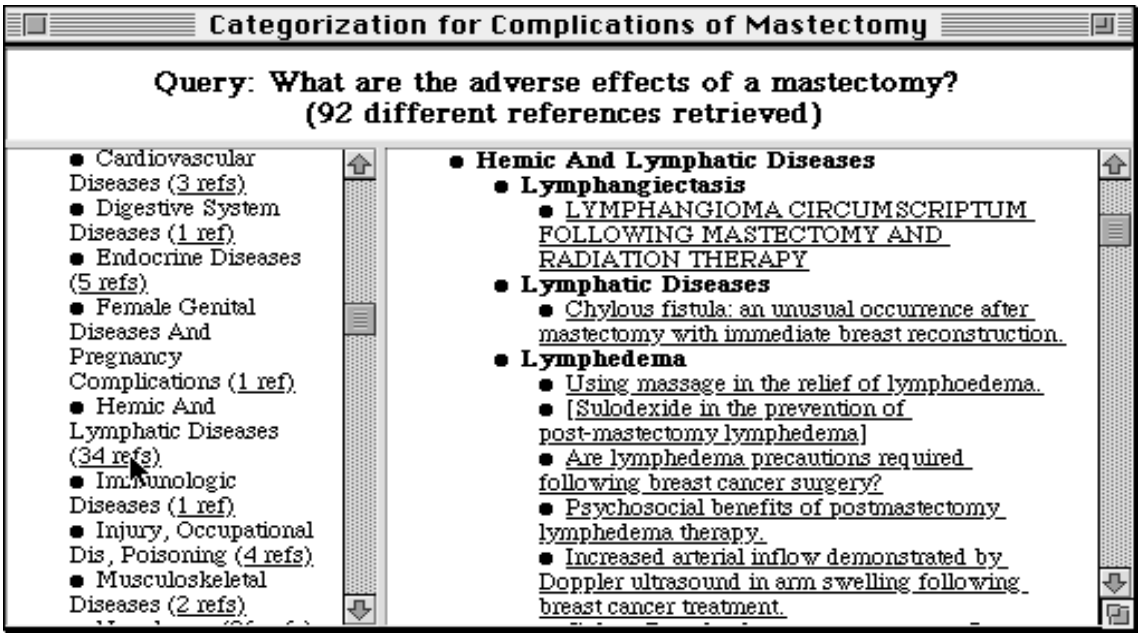


Figure 1.7 — DynaCat's display after clicking on the hyperlink corresponding the *Hemic And Lymphatic Diseases* category. The right window pane now shows the *Hemic And Lymphatic Diseases* category at the top of the screen. The user can click on any of the hyperlinks corresponding to the document's title, and that document's citation will appear in the right window pane.

1.8 Evaluation

In Section 1.6, I made two claims for my research: a usefulness claim, and a technical claim. I tested the validity of my two claims in separate evaluations.

1.8.1 Evaluation of Usefulness Claim

My usefulness claim is that the dynamic categorization of search results is more useful to users who have general questions than are the two other approaches to organizing search results: relevance ranking and clustering. A useful system helps users to

- Learn about the kinds of information that pertain to their query
- Find answers to their question efficiently and easily
- Feel satisfied with their search experience

I recruited patients and their family members from Stanford's Community Breast Health Project (CBHP 1997), the Stanford Health Library, and Stanford's Oncology Day Care Clinic. Every subject used all three organizational tools: (1) DynaCat, (2) a tool that clusters the search results, and (3) a tool that ranks the search results according to relevance criteria. Each subject used three different queries: *What are the ways to prevent breast cancer?*, *What are the prognostic factors for breast cancer?*, and *What are the treatments for breast cancer?* I randomized the query used with each tool and the order in which the subjects used the tools.

To measure the amount of information that the subjects learned using each tool, I asked each subject to list answers to the three queries before she used any tool, and to answer the same queries after she had used all the tools. For each tool, the amount that she learned was the number of new answers that she provided on the final answer list. The mean number of new answers was greater when subjects used DynaCat than when they used the cluster tool or the ranking tool; however, this difference was not significant. The tool used may have had an influence on the amount learned, but the number of new answers was correlated more strongly with how recently the subjects used a tool to find answers to that question, rather than which tool they used.

All subjects completed two types of timed tasks to determine how quickly they could find information related to the query. For the first type of timed task, subjects found as many answers as possible to the general question (e.g., *What are the preventive actions for breast cancer?*) in a 4-minute time limit. When the subjects used DynaCat, they found nearly twice as many answers as they did with the other two tools. This difference was significant ($p < 0.05$).

For the second type of timed task, I measured the time that it took the subjects to find answers to two specific questions (e.g., *Can diet be used in the prevention of breast cancer?*) that related to the original, general query. I found no significant difference among the tools. The time that it took subjects to read and understand the abstract, rather than the time that it took them to find a document among the search results, most heavily influenced the time for them to find an answer.

I used a questionnaire to assess many aspects of user satisfaction — for example, the clarity of the organization of search results, the ease of tool use, the usefulness of the organization, and the accuracy of the organization. This 26-question, user-satisfaction questionnaire is reproduced in Appendix A. On 13 of the 14 questions that requested quantitative answers, the satisfaction scores for DynaCat were significantly higher ($p < 0.05$) than they were with for either the ranking tool or for the cluster tool. On all the yes—no questions, DynaCat was rated with many more positive responses than was either the ranking tool or the cluster tool. No subjects chose DynaCat as the worst tool, and most of the subjects (70 percent) chose DynaCat as the best tool.

In summary, the results showed that DynaCat is a more useful organization tool than the cluster tool or the ranking tool. DynaCat was significantly better than the other two tools in terms both of efficiency in finding answers to their original queries and of user satisfaction. The objective results for the amount learned were inconclusive; however, most subjects (87 percent) thought that DynaCat helped them to learn about the topic of the query, where only 47 percent thought that about the cluster tool and only 60 percent thought it about the ranking tool.

1.8.2 Evaluation of Technical Claim

My technical claim is that DynaCat meets the criteria defined in Section 1.6.1. To test this claim, I recruited physicians from Stanford University to assign documents to both DynaCat-generated categories, and clusters. I presented each subject with the query that the search engine used to generate the list of search results, the abstract and complete citation for each document in the search results, the list of categories that DynaCat selected, and the list of clusters that the cluster tool generated. For each citation, I asked the subjects (1) to read the document's title and abstract, (2) to list as many categories as appropriate that both described that document and answered the query, and (3) to identify one cluster that described the document. After the subjects completed those tasks, they filled out a questionnaire that asked about how meaningful the labels were, how well the groups corresponded to the search results, and how well the groups corresponded to the query.

I used the kappa statistic to measure the consistency among the subjects, and between the subjects and the document-organization tool. The consistency among the subjects ranged from fair to moderate. The consistency between the subjects and DynaCat fell within the same range, as did that for the consistency between the subjects and the cluster tool.

Subjects rated the categories higher than the clusters in terms of how meaningful the labels were, how well the groups corresponded to the query, and how well the groups corresponded to the search results, but none of these differences were statistically significant.

In summary, the technical evaluation demonstrated that the categorization generated by DynaCat was about as consistent with the physicians' categorizations as the physicians's categorizations were with each other. These results suggest that DynaCat creates reasonable document categories and assigns documents to categories appropriately.

1.9 Guide for the Reader

Chapter 2 reviews previous research in organizing documents and presents the common schemes for representing documents and queries in information-retrieval systems. I introduce the three approaches to organizing documents: relevance ranking, clustering, and classification. I detail their algorithms, their uses in the various stages of the search process, and their advantages and disadvantages in organizing search results.

Chapter 3 describes the components of dynamic categorization. I describe the domain-specific knowledge that is in the form of two domain models: a terminology model, and a query model. I present the system architecture, and specify each component.

Chapter 4 presents the usefulness evaluation. I describe the study and report the results.

Chapter 5 details the evaluation of my technical claim. I explain the study design and state the results.

Chapter 6 summarizes the contributions of my research, the limitations of my approach, and the possibilities for building on my research in the future.

Chapter 2

Previous Approaches to Organizing Documents

Most search systems organize documents returned from a search into an ordered list called a relevance-ranked list. In Section 2.3, I discuss how systems generate relevance-ranked lists, and how those lists can be used to present the search results to the user.

Search systems also can organize documents into groups. In Sections 2.4 and 2.5, I present the main approaches to organizing documents into groups: categorization and clustering. For each approach, I describe how it works, and how it is used in the different steps of the search process, emphasizing how it is used to present search results to the user.

All approaches to organizing documents are based upon a document representation and optionally a query representation. I discuss the common document and query representation schemes in Section 2.1 and Section 2.2. Most of these representations were developed to improve the precision and recall of search systems, rather than to improve the organization of the search results. However, systems that organize documents use many of these same representation schemes.

2.1 Document Representations

Search systems can use a wide variety of document representations. These representations try to satisfy two goals: (1) to enable search systems to find all of documents that are relevant to the user's query without returning irrelevant documents—measured in terms of precision and recall, and (2) to find the relevant documents quickly.

2.1.1 Vector Space

Nearly all full-text search systems represent documents using the **vector-space** paradigm, where each document in the collection is represented by a vector of terms that occur in that document (Salton, Wong, et al. 1975; Salton and McGill 1983; Salton 1989). The *i*th element in a document's vector represents the value of the *i*th term in that document. Each document's vector acts as the coordinates for that document in a multidimensional term space. This paradigm provides an intuitive way for viewing documents as positions in space, where the similarity between two documents is the distance between them.

Systems usually preprocess the documents before creating the final term vector. Most search systems first remove **stop words**: common words, such as prepositions, conjunctions, and articles that do not influence the meaning of the documents. For example, the words *a*, *is*, and *of* are typically considered stop words and would be removed from the vector representation of a document.

Some systems also replace each word by its morphological root form. A simple form of morphological processing is **stemming**, which consists of replacing plural nouns (such as *complications*) with their singular form (such as *complication*), and stripping suffixes from root words. For example, the words *quickly* and *quicker* would be stripped to their root word *quick*. In more sophisticated morphological analysis, inflectional verb forms (such as *ate*, *eaten*, *eating*) could be replaced with their infinitive form (such as *eat*). Such processing is meant to provide better recall because a search on one term would match every instance of the term's morphological variants that occur in the text. However, such processing can conflate multiple terms with extremely different meanings. For example,

the verb *parking* would be stemmed to *park*, which would be indistinguishable from uses of the noun *park*. This problem is prevalent particularly in medicine where the suffix conveys a significant part of the meaning of a word. Consider an example from (Purcell 1996), the term *sinus* denotes a cavity within a bone, the term *sinusitis* denotes an inflammation of a sinus within the skull, and *sinusoid* denotes a large diameter capillary. All of these terms would be stemmed to the same root word *sinus* even though the distinctions in their meaning is important. Studies have shown that stemming algorithms in medical information-retrieval systems can reduce the search precision dramatically (Hersh and Greenes 1990).

After the preprocessing, each document is represented by a vector of all the remaining document terms that occur within the document collection. If the document collection contains n unique terms, then each document is represented as a vector of length n: (t_1, t_2, t_3, . . ., t_n) where t_i has a value of 0 if term i is absent in the document, and has a positive value if term i is present. The exact positive value depends on which **term-weighting** scheme the search engine uses. For the simplest case of no term weighting, the value is 1 if the term is present.

Researchers have proposed and experimented with a large number of term-weighting schemes (Robertson and Walker 1994; Kwok 1996; Salton, Yang, et al. 1975; Salton and Buckley 1988). The most successful schemes use some form of **term frequency times inverse document frequency (tf-idf),** where **term frequency (tf)** is the number of occurrences of the term in the document that is being represented, and **inverse document frequency (idf)** is the inverse of the number of occurrences of the term in all documents in the collection. The intuition behind term frequency is that terms that occur frequently in a document are more likely to indicate a central topic of that document than are terms that

occur rarely in the document. Inverse document frequency is used to indicate how well the term distinguishes among the documents. One common tf-idf term-weighting scheme is

$$w_i = tf_i \times \log \frac{N}{df_i}, \qquad \textit{tf-idf}$$

where w_i is the weight of term i,
tf_i is the frequency of term i in the document,
df_i is the frequency of term i in the entire document collection, and
N is the total number of documents.

As a variation on this basic scheme, some systems adjust the term weight depending on the portion of the document in which the term occurs. For example, the retrieval system based on the Hepatitis Knowledge Base (Bernstein and Williamson 1984) calculates term weights based on the number of times that the term occurs within an entire collection, on the number of times that the term occurs within a given document, and on the structural position of the term within the document, such as within summary paragraphs versus within text paragraphs. Other systems, such as the Sandwich Interactive Browsing and Ranking Information System (SIBRIS) (Wade, Willett, et al. 1989), weight terms that appear in the title more heavily than terms that appear in only the text for the document. Another variation weights the terms based on their proximity (Keen 1991; Robertson, Walker, et al. 1994).

2.1.2 Controlled Vocabulary

Some search systems represent documents using only terms from a **controlled vocabulary**, a predefined set of allowable terms. The Semantic And Probabilistic Heuristic Information Retrieval Environment (SAPHIRE) is one example of such an information-retrieval system for the domain of medicine (Hersh and Greenes 1989). It maps words and phrases in a document to a set of canonical terms. The list of canonical terms is derived from the UMLS Metathesaurus, an extensive medical terminology model (McCray, et al. 1993). SAPHIRE represents each document by a weighted vector of these canonical terms. The Knowledge Server (now called the Knowledge Authority) by Lexical Technologies is another example of a medical information-retrieval system that maps words and

phrases to canonical terms, but it does not weight the term vector with any frequency information (Tuttle, Sherertz, et al. 1994). A failure analysis of several studies that compared controlled-vocabulary systems with the traditional vector-space systems found that the performance of search systems using the controlled-vocabulary representation depends largely on the quality of the terminology model (Hersh and Hickam 1992; Hersh and Hickam 1993).

2.1.3 Structured Documents

Documents can be structured into their different syntactic components as described in Section 2.1.3.1 or they can be broken into different semantic components as described in Section 2.1.3.2 and Section 2.1.3.3.

2.1.3.1 Document Components

Many bibliographic search systems break a document into its syntactic components such as title, authors, journal, keywords, and abstract. As in the vector-space model, the document is represented by the words occurring in the document; however, each component has its own vector of words. This separation allows the user to search explicitly for the occurrences of words within a particular component such as the title. Web documents also contain syntactic components such as the title, URL, headers, and hyperlinks. In principle, web search systems could allow users to search for words within these syntactic components, but in practice, few systems allow the user to specify syntactic components in the search process. Most web search systems do not publicize their term-weighting or ranking algorithms; however, they probably weight words within headers and metatags higher than words that occur in other sections of the web page.

An alternative approach is to represent explicitly the semantic components of a document, rather than only its syntactic components. The RiboWEB system uses this approach to represent scientific publications about the ribosome (Altman, Bada, et al. 1999; Bada and Altman 1999). It contains a large knowledge base of relevant published data, and a suite of computational modules capable of processing this data to test hypotheses about the struc-

ture of the ribosome. This explicit representation of the published data allows users to perform a variety of computational functions on this data, and compare results in a way that is not possible with pure-textual representations of the literature. Sim has proposed a similar approach for representing clinical trial information in a central repository (Sim 1997).

2.1.3.2 Structured Abstracts

Structured abstracts impose a semantic structure and a specific format to a document's abstract. Many journals have adopted structured abstracts with the hope of facilitating peer review, helping the reader access the document, and improving electronic searching (Evans 1993; Huth 1987; Lilleyman and Lowe 1992; Lock 1988; Squires, et al. 1992). Unfortunately, several problems with structured abstracts have limited their ability to meet those objectives. First, each journal has its own guidelines and format for the structure. These varying structures make it difficult for search systems to provide uniform access to the structure when searching across multiple journals. This variety also makes it more difficult for the reader to learn about and remember the meanings of the different components of the structured abstract. In addition, some journals have rejected structured abstracts because the imposed structure could inhibit the expressiveness and creativity of authors (Heller 1991).

2.1.3.3 Context Models

Context models provide a semantic structure to full-text documents without imposing specific format guidelines (Purcell 1996; Purcell, et al. 1997). Someone must assign each sentence in a document to one or more **contexts** that describe the semantic theme of that sentence. For example, a search system could distinguish between a document that contains the term *breast cancer* in the context of the eligibility criteria of a study from a document that mentions *breast cancer* in the context of the adverse effects of an intervention. The user can then provide her query by specifying terms and the context in which those terms should appear. Purcell has developed context models for clinical research articles, case reports, and review articles in the medical literature. In an evaluation of her system at a fixed level of recall, she demonstrated that searches using the context models resulted in

slightly better precision than the same searches with a Boolean, full-text search system. Contexts are assigned manually to individual sentences; thus, they have been used in only a research setting, although several journals are considering adapting them into their editorial process.

2.2 Query Representation

Search systems use a variety of techniques for representing queries. In Sections 2.2.1 through 2.2.4, I describe Boolean, vector-space, and natural-language query representations as well as queries in the form of documents. Queries in any of those forms can be used to generate search results that are organized in some way, but many systems rely on a vector-space representation of both the documents and the queries.

2.2.1 Boolean Queries

Boolean queries are combinations of terms using the operators *and*, *or*, and *not*. They indicate which terms should be present or absent in the retrieved documents. For example, if the user issued a query of *mastectomy and (cancer or neoplasm)*, the search system would return all documents that contain both the words *mastectomy* and *cancer*, both the words *mastectomy* and *neoplasm*, or all three words—*mastectomy*, *cancer*, and *neoplasm*. Some systems allow Boolean queries over the different document components such as author, journal, title, keywords, abstract. Many bibliographic search tools such as Inspec and Melvyl MEDLINE support only Boolean queries. Many full-text retrieval systems allow Boolean queries as an advanced search option. However, studies have shown that searchers often confuse and misuse the Boolean operators (Borgman 1986); thus, most full-text systems support vector-space queries as their default or naive user interfaces.

2.2.2 Vector-Space Queries

The vector-space query representation is the same as the vector-space document representation discussed in Section 2.1.1; a query is represented as a vector of terms that occur in the query (Salton, Wong, et al. 1975; Salton and McGill 1983; Salton 1989). A searcher enters a query as a list of terms that should be in the retrieved documents. This query-formulation process is often much easier for users than entering a query as a Boolean expression. This advantage is one reason why most web-based search systems and other full-text retrieval systems represent queries as vectors of terms. This representation is also used by most systems that rank the retrieved documents as described in Section 2.3.

2.2.3 Natural-Language Queries

Most systems that allow natural-language queries transform those queries into a vector-space representation in the same manner as described in Section 2.1.1 for documents. This technique provides consistent processing of both the documents and the queries; however, it also may result in a mismatch between the searchers' expectations and the system's capabilities. If users are allowed to enter their query in natural language, they may assume that the system uses natural-language processing techniques to understand both the query and the content of the documents, and thus raise the expectations of the searcher. Unlike systems that transform users' queries into a vector of words, some systems do use natural-language processing techniques in matching documents with queries. I describe a particular technique called information extraction in more depth in Section 3.3.3.2.

2.2.4 Documents as Queries

Even when users are allowed to express their information need using natural language, they may have difficulty articulating that need exactly. Users may find it much easier to identify a document that is close to meeting their needs, and thus some search systems allow documents as queries. This representation allows the user to provide an exemplar for the type of documents that she would like retrieved. The query document is trans-

formed into a vector and matched against the vector-space representation of the documents in the document collection. Many systems such as PubMed (NLM 1997c), employ this technique in the query reformulation stage; after the search system returns documents from the initial query, the user can select one of those retrieved documents and tell the search system to find similar documents. This process is a form of **relevance feedback**, where users provide feedback to the search system about which documents are relevant to their query. Studies have shown that relevance feedback does improve system performance, in terms of the precision-recall metrics (Buckley, Salton, et al. 1994).

2.3 Relevance Ranking

The purpose of **relevance ranking** is to order the documents returned from a search according to their estimated relevance to the query. One simple form of organization that could be considered a relevance ranking is listing the documents in reverse chronological order, which is the approach taken by many Boolean information retrieval systems. However, most vector-space systems determine a document's rank based on some measure of how well the document matches the query. This measurement is called a similarity score. I describe some of the common techniques for calculating a similarity score in Section 2.3.1.

2.3.1 Similarity Scoring

In ranking the documents for a given query, the search system computes a **similarity score** between the query and a document. The documents are ranked from the highest to lowest similarity score. Most of these similarity computations rely on a vector-space document and query representation, where the similarity score is a measure of the distance between the query and the document in the multidimensional vector space. Researchers have explored a variety of similarity scoring algorithms (Belkin and Croft 1987; Salton and Buckley 1988; Salton 1989). All these algorithms can be used to calculate the similarity between documents in addition to that between a query and a document. The simplest

similarity scoring algorithm, adds the weights of the terms that both objects have in common. However, this metric does not account for the varying length of documents; long documents would usually have a greater sum than short documents. Most tests have shown that similarity metrics that are normalized by document length produce better results. Common, normalized similarity metrics include Dice's coefficient, Jaccard's coefficient, the cosine coefficient, and the overlap coefficient (van Rijsbergen 1979):

$$\sum_{i=1}^{n} (X_i \times Y_i), \qquad \textit{Simple matching coefficient}$$

$$\frac{2\sum_{i=1}^{n} (X_i \times Y_i)}{\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n} Y_i^2}, \qquad \textit{Dice's coefficient}$$

$$\frac{\sum_{i=1}^{n} (X_i \times Y_i)}{\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n} Y_i^2 - \sum_{i=1}^{n} (X_i \times Y_i)}, \qquad \textit{Jaccard's coefficient}$$

$$\frac{\sum_{i=1}^{n} (X_i \times Y_i)}{\sqrt{\sum_{i} X_i^2 \sum_{i} Y_i^2}}, \qquad \textit{Cosine coefficient}$$

$$\frac{\sum_{i=1}^{n} (X_i \times Y_i)}{\min(\sum_{i=1}^{n} X_i, \sum_{i=1}^{n} Y_i)}. \qquad \textit{Overlap coefficient}$$

Where, X is the vector representation of a document.

Y is the vector representation of a query or another document.

X_i is the weight of term i in vector X.

Y_i is the weight of term i in vector Y.

n is the total number of terms in the document collection.

Many of the commercial search tools use an undisclosed formula for calculating the term weights, the similarity score, and thus the ultimate relevance-ranking criteria.

2.3.2 Use in Presentation of Search Results

Many interfaces for relevance ranking consist of only the ordered list of search results, possibly with the relevance score displayed next to each document's title or summary. Usually, the interfaces do not indicate the criteria used to generate the relevance score. One exception is the Lycos Pro web-based search engine (Lycos 1997), which allows the user to designate the importance of six factors (matching every query word, frequency of query words, appearance of query words in the title, appearance of query words early in the text, appearance of the query words close to each other, and appearance of the query words in the exact order that was specified) in calculating the relevance score (see Figure 2.1). This interface gives the user some control over the ranking. However, the interface to the search results does not indicate how each document fared on each of the six factors; it shows only the combined relevance ranking (see Figure 2.2). For example, the user cannot determine whether the top-ranked document received that rank because it was the only document that contains both words in the query, or whether many of the documents contain both words but the first document met other important ranking criteria as well.

TileBars (Hearst 1995) is one of the few interfaces that shows how the document and portions of the document are related to the query terms (see Figure 2.3). The user enters each topic (a topic may be one or more words) of her query on a different line. The interface graphically shows the user the relative length of each document in the search results, the

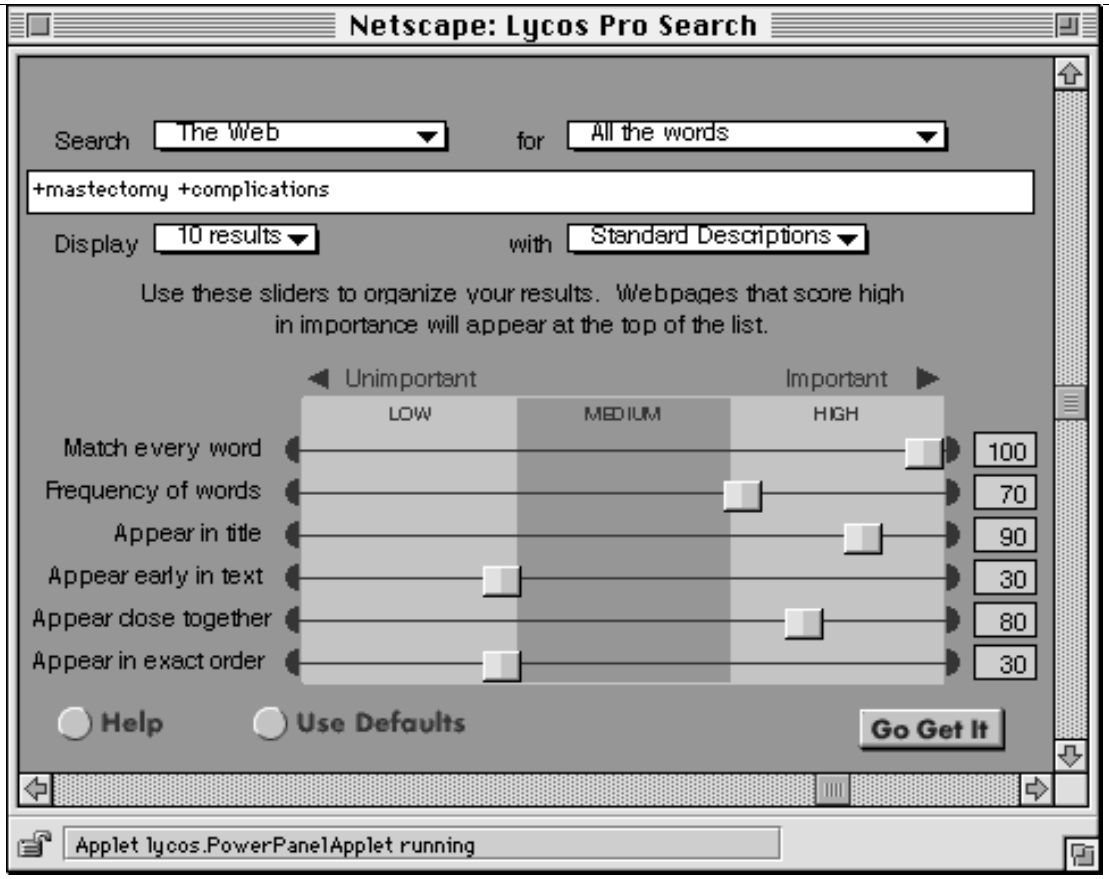


Figure 2.1 — Interface that allows the user to adjust relevance-ranking criteria for the Lycos Pro web-based search engine (Lycos 1997).

frequency of the topic words in each document, the distribution of the topic words within the segments of each document, and the distribution of the topics words in relation to the distribution of the other topic words.

2.3.3 Advantages and Disadvantages of Relevance Ranking

Relevance ranking may be most useful when the user has a specific question, and when only a few documents are of interest to her. If the system places those documents near the top of the search results, the user should have little difficulty locating them. However, the relevance ranking may not accurately reflect an individual user's relevance judgements. The relevance of a document for a particular user depends on many factors that may not captured by the search tool. A search tool has little or no information about the user or the

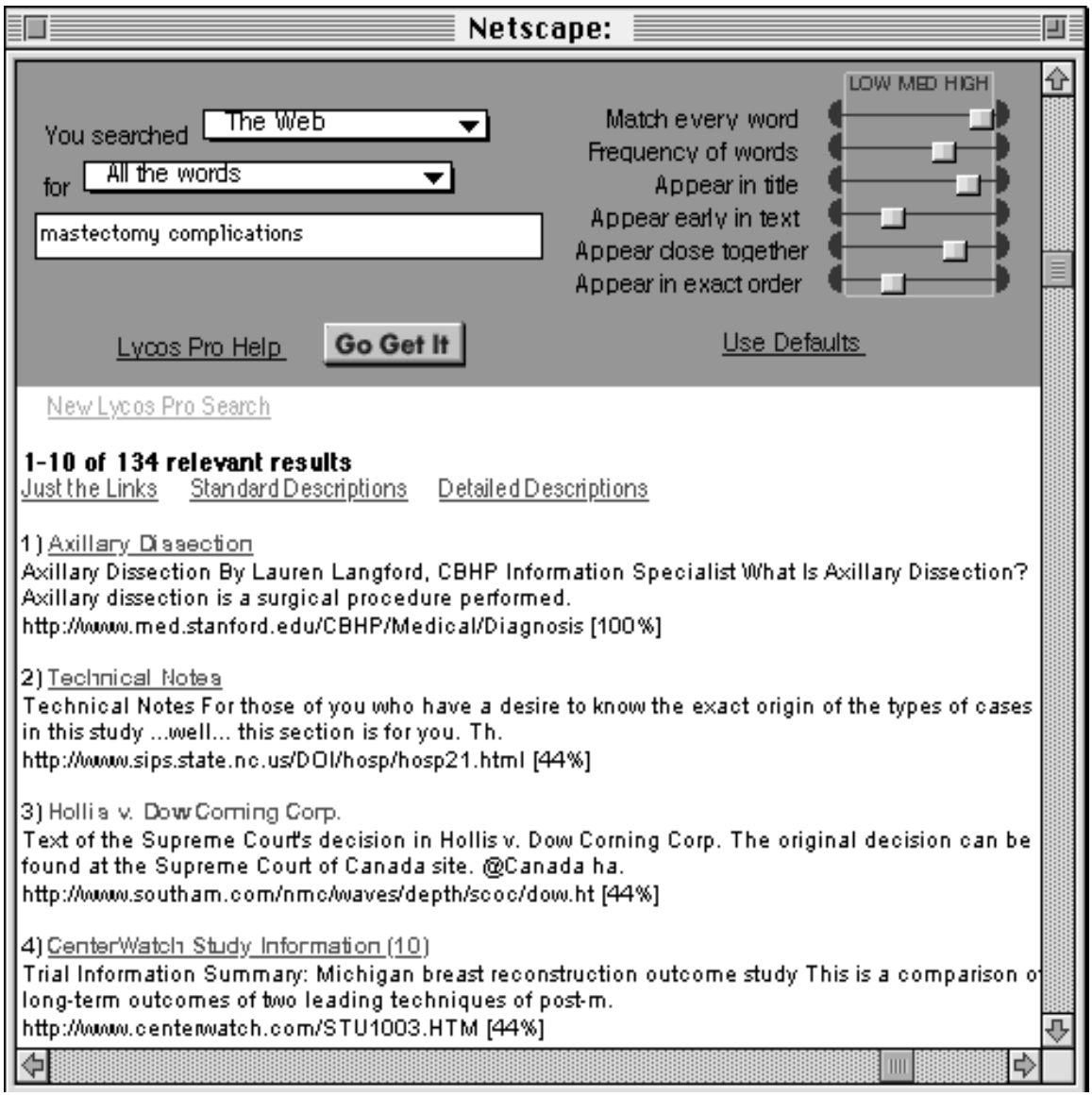


Figure 2.2 — Lycos relevance-ranking interface. This interface ranks the search results using the criteria in the upper right of the screen. The numbers in brackets indicate the relevance score for the document.

context of her query. It does not know what she already understands regarding the topic of the query, what documents she has already read, why she is asking the query, or how she wants to use the documents that she finds. Without such information, it is difficult to assess the relevance of a document to a query. Search tools could attempt to elicit such information from the user, but this process could be time consuming and annoying to the

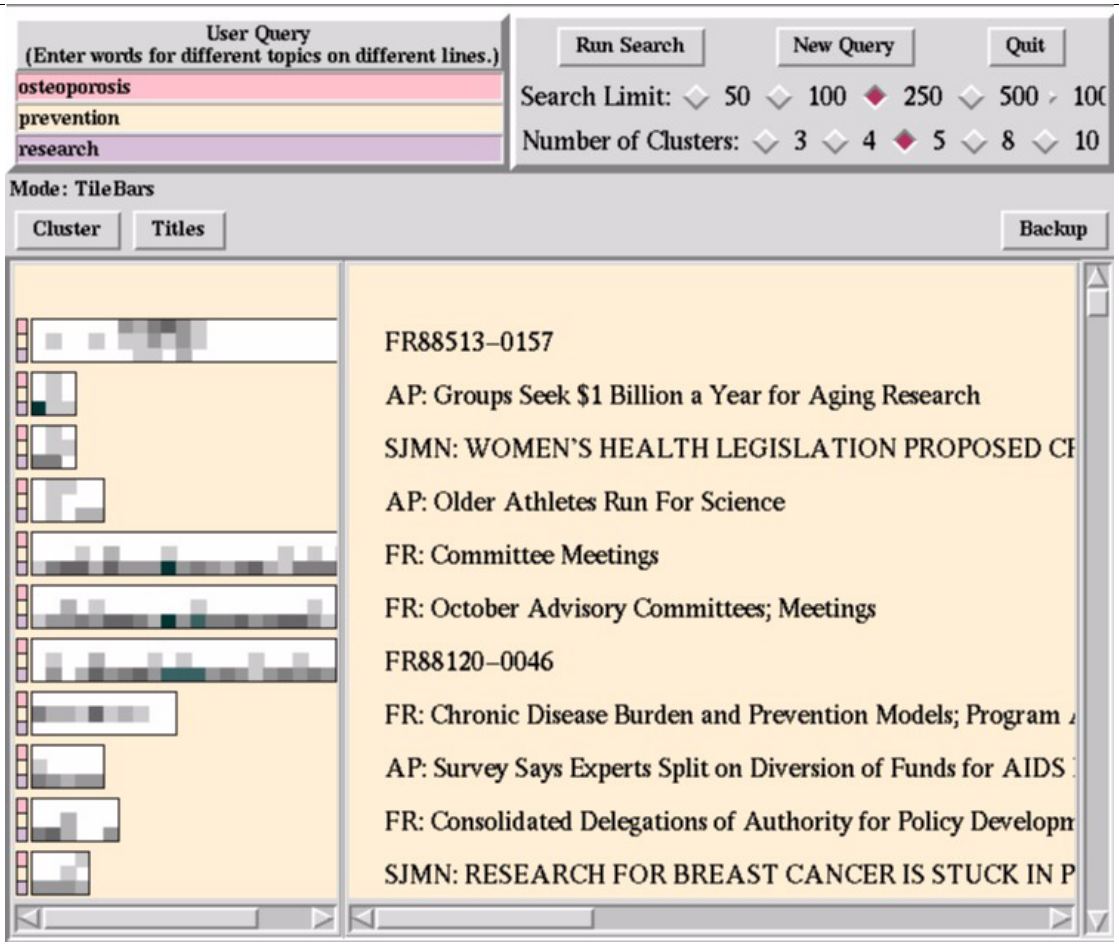


Figure 2.3 — TileBars interface[a]. At the top part of the screen, the user enters her query as a set of topics where the word(s) for each topic is entered on a different line. The tool displays the search results at the bottom right part of the screen with its corresponding TileBars appearing in large rectangles on the left. Each row corresponds to a query topic (in the same order as in the query), and each column of small squares represents a segment within the document. For example, the TileBar corresponding to the first document shows that the words from all three topics (*osteoporosis*, *prevention*, *research*) appear in the document. It also shows that, toward the middle of the document, there are three segments in which all three topics appear, and these are the only segments where the word *research* appears. The shading indicates the frequency of occurrence such that a darker shaded segment indicates that the topic word(s) appear more frequently in that segment than one with a lighter shade.

a. This figure appears courtesy of Marti A. Hearst.

user. It may be more efficient to present the results in a way that allows the user to assess and choose easily which results are most important given her situation.

When the user has a broad query, relevance ranking is unlikely to be useful. In such cases, many documents are relevant to the user's query; looking at each document is too time

consuming. Interfaces such as TileBars can help users to see where and how frequently the query terms occur in the documents, but they do not help the user to understand the kinds of information represented in the search results. Such information is not visible in relevance-ranked lists of search results. One possible solution to this problem is to present the user with a summary of the results that provides enough information for her to decide which documents to examine more closely. One way to provide such a summary is to group the search results according to the contents of the documents, such as through document clustering, document classification, or dynamic categorization.

2.4 Document Clustering

In this section, I present a basic overview of document clustering. I discuss a few variations in the algorithms; however, I do not present a comprehensive review of all clustering algorithms, or even of all algorithms that have been used to cluster documents. Many other publications contain detailed reviews of document clustering algorithms (Rasmussen 1992; van Rijsbergen 1979; Willett 1988).

Researchers have applied three kinds of clustering to improve the search process: term clustering, citation-based clustering, and document clustering. **Term clustering** groups the terms in a document collection based on the documents in which they co-occur. Search tools use this kind of clustering to help users reformulate their queries by displaying clusters of terms that users may want to add or exclude. **Citation-based clustering** groups documents based on the citations that they share. This technique is used to help people find connections and trends in the literature of a field (Small and Sweeney 1985). **Document clustering** groups documents based on their content. In this document, I describe only this third kind of clustering. My goal is to provide enough information for understanding the basic steps in document clustering, the advantages and disadvantages of document-clustering approaches, and the differences between document clustering and dynamic categorization.

2.4.1 Document-Clustering Algorithms

Like relevance-ranking algorithms, clustering algorithms represent documents as vectors of terms (see Section 2.1.1). Document clustering systems use an unsupervised algorithm to create the clusters. They take documents as input, extract or select the features of the documents, and form clusters based on a calculation of similarity between individual documents or between an individual document and a representation of the clusters formed so far. The similarity calculation, described in Section 2.3.1, is based on only the selected features for that document collection. The feature-selection process is described in Section 2.4.2.

Clustering algorithms can be either **hierarchical** and form a tree-like organization of documents, or they can be **nonhierarchical** and form a flat set of document groups. Most nonhierarchical clustering algorithms group documents into a preset number of clusters, *K*, which could be supplied by the user or generated by the clustering system. First, either the user or the clustering system chooses *K* seeds to represent the centers of the *K* clusters that the system will create. For each document, the system uses a similarity metric to calculate the similarity between each seed and that document (see Section 2.3.1). The document is assigned to the cluster with the most similar seed. The process is repeated for each document. An advantage to nonhierarchical methods is that they are computationally more efficient than are hierarchical methods: nonhierarchical clustering is $O(KN)$ where N is the number of documents, as opposed to $O(N^2)$ for hierarchical clustering. A disadvantage is that both the number and the centers of clusters must be determined a priori.

Hierarchical clustering algorithms can be either **divisive**, where all documents start out in one cluster and are broken into many, or **agglomerative**, where all documents start out as their own clusters and are grouped pairwise into a smaller number of clusters. Most of the work on document clustering has concentrated on the hierarchical agglomerative clustering methods, such as single linkage, complete linkage, group average, and Ward hierar-

chic agglomerative clustering methods (Rasmussen 1992). All the hierarchical agglomerative methods follow the same basic algorithm:

1. Identify the two closest clusters (using a similarity metric), and combine them into one cluster.
2. Repeat step 1 until only one cluster remains.

No single algorithm has been shown to produce uniformly better clusters, but single-link techniques have been shown to create consistently poor clusters (Voorhees 1985, Willett 1988).

2.4.2 Feature Selection

Clustering algorithms represent documents as a set of **features** that they use to determine how to group the documents. The selection of features is important because these features are the only information about the documents that the clustering algorithms have. Many systems use some subset of the terms in the document collection as the document features. The problem with using all the terms is that the efficiency of both clustering and classification algorithms depends on the number of features used. Because there are many terms in even a small document collection (at least 10^5 terms), efficiency can be a major problem (Sahami, et al. 1998). Thus, researchers have focused mainly on techniques for reducing the number of features (terms) used in representing the document. As a first step, nearly all approaches remove stop words (see Section 2.1) and punctuation from the features. Most approaches to feature selection eliminate all terms whose frequency is above a certain maximal threshold or below a certain minimal threshold. Some approaches set those thresholds arbitrarily, whereas others base the thresholds on some principle such as Zipf's law (van Rijsbergen 1979), entropy, or information theory (Koller and Sahami 1996). Another approach is to transform the space of features into a reduced set of features by finding relationships among terms in the collection. Both latent semantic indexing (Deerwester, et al. 1990, Dumais 1993) and linear least-squares fit (Yang and Chute 1994b) are techniques that have been used for such feature-space transformations.

2.4.3 Use in Matching Documents to Queries

Most systems that employ clustering techniques group all documents in a collection in an effort to improve either the efficiency or effectiveness of retrieving relevant documents for a given query. These approaches are based on the **cluster hypothesis**, which states that closely associated documents tend to be relevant to the same requests (van Rijsbergen 1979).

To increase the efficiency of finding documents, search systems match the query against each cluster representation, rather than against each document representation. Only documents that belong to matching clusters are assumed to be relevant to the query. Search tools use this process to accelerate the traditional vector-based searching, but more efficient algorithms for vector-based searching have nearly eliminated this use of clustering (Rasmussen 1992).

Another goal in using clustering is to increase search effectiveness by improving the system's recall. Search systems used clustering to broaden a search request. However, several studies have shown that cluster-based searching is no more effective than, and sometimes is less effective than, typical vector-based searching (Griffiths, et al. 1986, Jardine and van Rijsbergen 1971, Rijsbergen and Croft 1975).

2.4.4 Use in Presentation of Search Results

Recently, a few researchers have applied clustering techniques to help users navigate, and gain a high-level understanding of, an entire document collection, or of the results of a search. For this type of clustering application, the systems also need a way to describe each cluster to the user. They usually label each cluster using the highest-weighted terms from the center of that cluster. For efficiency reasons, these approaches typically use nonhierarchical clustering methods.

Scatter/Gather is a document-clustering system that first was applied to help users understand the topics of a document collection as a whole (Cutting, et al. 1992). In the past cou-

ple of years, researchers have been using Scatter/Gather to cluster search results (Hearst, et al. 1995). Scatter/Gather is an interactive tool that allows the user to select clusters of interest, and to recluster only the documents in those cluster. Figure 2.4 shows an example of the Scatter/Gather interface.

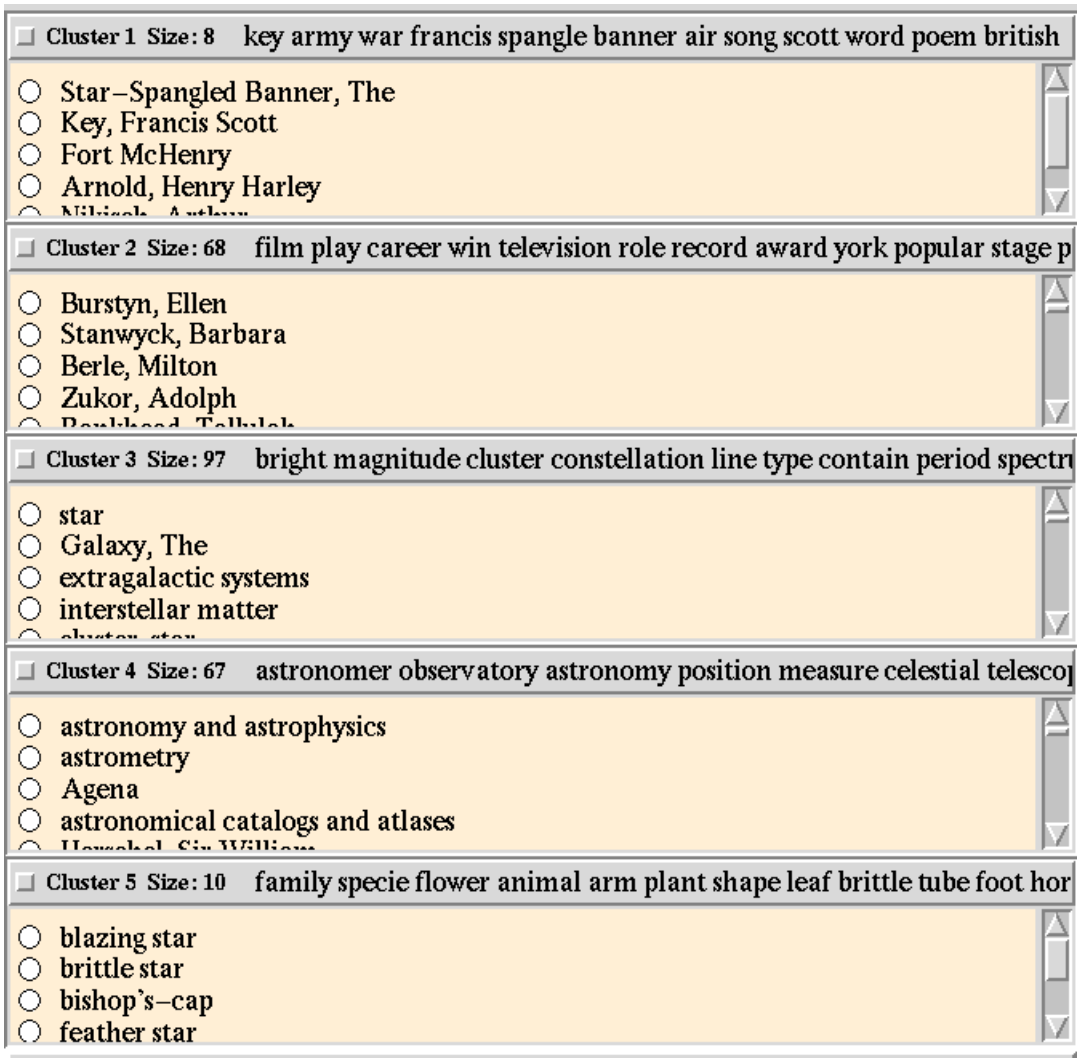


Figure 2.4 — Scatter/Gather interface[a]. This screen shows the 5 clusters that Scatter/Gather created from a search on encyclopedia articles that contain the word *star*. Each cluster is displayed in its own section of the screen. Each section is labeled with the cluster number, the number of documents in that cluster, and the representative words from that cluster. The section also contains a scrollable list of the titles of the documents that belong to that cluster. In this example, the clusters appear to correspond to the different senses of the word *star* (e.g. famous person, the celestial body).

a. This figure appears courtesy of Marti A. Hearst.

The Service for Organizing Networked Information Autonomously (**SONIA**) (Sahami, et al. 1997a; Sahami 1998) is a prototype service in the Stanford Digital Libraries Testbed that provides document clustering as part of the SenseMaker interface (Baldonado and Winograd 1997; Baldonado 1997). Through SenseMaker, users can query a number of information sources, and then use the feature *bundling by similar content* to employ SONIA for document clustering. SONIA first stems each term in the document, sends the vectors through a feature-selection process, and uses either of the nonhierarchical algorithms K-Means (Krishnaiah and Kanal 1982) or AutoClass (Cheeseman, et al. 1988) to cluster the documents.

Other researchers have displayed document clusters using specialized graphics such as dendrograms (Allen, et al. 1993), or starfields (Allan and Hirsch 1997), but such displays do not label the clusters and may not help the user to understand the groupings. One study that compared clustering systems found that a system that showed the textual information associated with the documents was more useful than the other systems that showed the clusters as projections onto a 2D or 3D space (Kleiboemer, et al. 1996).

2.4.5 Advantages and Disadvantages of Document Clustering

As discussed in (Hearst to appear), clustering has usability tradeoffs when applied to organizing search results. The main advantage of clustering is that it may reveal previously hidden but meaningful themes among documents. Such themes correspond to those found in the search results, rather than being predefined. Since clustering is an unsupervised approach, it also requires no domain-specific knowledge.

Most of the disadvantages of clustering stem from its unsupervised nature. The generated clusters indicate associations among the documents in the cluster; however, those associations may not be meaningful to the user. Clustering systems also have no clear way to convey the meaning of the clusters. Although most systems display labels of representative terms in the cluster, the user may not be able to determine the meaning of such a list of

terms. Finally, the clusters are based solely on the search results, even though only a subset of the possible clusters may be of interest to the user for any given query.

2.5 Document Classification

Document classification is a method (manual or automatic) to assign documents to labeled categories that represent themes discussed in those documents. In Section 2.5.1, I describe the common algorithms for automatic document classification. I describe how such categories can be used in the presentation of search results (Section 2.5.2), and I summarize the advantages and disadvantages of document classification (Section 2.5.3).

2.5.1 Automated Algorithms

Previous approaches to automatic document classification have used a wide variety of supervised-learning algorithms, including decision-rule induction (Apte, et al. 1994), decision-tree induction (Lewis and Ringuette 1994; Tong and Appelbaum 1994), nearest neighbor algorithms (Massand, et al. 1992; Yang and Chute 1994b), Bayesian classifiers (Lewis and Ringuette 1994; Lewis 1992b), discriminant analysis (Hull 1994), and neural networks (Ng, et al. 1997; Wiener, et al. 1995). Systems that use these approaches follow the same basic steps for categorizing documents. They require a training set of documents, where each document is assigned to any number of predefined categories. Each document in the training set is represented as a vector of terms (as described in Section 2.1), and some feature-selection process (see Sections 2.4.2 and 2.5.1.1) is applied to produce the final feature vector. For each predefined category, the goal is to determine a function of the features in the feature vector that accurately predicts whether or not a document belongs to the category.

2.5.1.1 Feature Selection for Supervised-Learning Techniques

Classification algorithms can apply the same feature-selection approaches as clustering algorithms (see Section 2.4.2); however a few feature-selection algorithms can be used in only document classification techniques because they are based on the training set.

One approach creates a local dictionary for each category (Apte, et al. 1994). The terms in the dictionary are only those terms that are present in the documents from the training set that were assigned to the category. These terms are the only ones used to determine whether a new document belongs to the category.

2.5.2 Use in Presentation of Search Results

Most of the interfaces that use categories to display search results take advantage of manually assigned category labels. Both the Cat-a-Cone (Hearst and Karadi 1997) system and Yahoo! use such categories in their display of search results.

The Cat-a-Cone interface integrates search and browsing of large category hierarchies with their associated text collections. One central feature of this interface is that the category labels are displayed separately from the documents. A ConeTree (Card, Robertson, et al. 1996) displays category labels, and a WebBook (Robertson, Card, et al. 1993) shows retrieval results. The left-hand page shows the title and the category labels associated with the document. The right-hand page shows the abstract associated with the document. Books that are the results of previous searches are stored in the workspace on the bookshelf, and thus can act as a memory aid for the user.

Although this interface might help the user to understand the content of each document individually, it might not help the user to see a summary of the overall content of the search results. The user can flip through the pages of the book, and observe the changes in the ConeTree. However, the interface shows the category hierarchy for each document individually, rather than showing the category hierarchy for the entire set of documents in the search results.

In contrast, the Yahoo interface shows the user's search results in an alphabetic listing of the categories. These categories help the user gain some knowledge of the content of the search results. However, when many categories are listed, the user may be just as overwhelmed by the long list of categories as she was by the long list of search results.

2.5.3 Advantages and Disadvantages of Document Classification

The advantage of classification systems over clustering systems is that the classification systems provide meaningful labels and groupings of the documents, but clustering systems do not. However, these labels must be predefined. If a theme for which there is no label is discussed in the search results, classification systems have no way to detect and label that theme. If most search results fall into one category, classification systems cannot divide a category to illustrate subthemes. When documents are assigned to multiple categories, many of those categories may be irrelevant to any given query.

2.6 Summary and Comparison to Dynamic Categorization

The document representation is fundamentally different in dynamic categorization compared to other approaches. Dynamic categorization uses a semantic-based representation of the terms in the documents; it incorporates the semantic type of each term, rather than basing the representation solely on the presence of terms. Most other approaches to document organization (relevance ranking, clustering, automatic document classification) represent documents solely by the occurrences of specific terms.

As opposed to relevance ranking, classification, or clustering, dynamic categorization exhibits all four desirable characteristics (see Section 1.4):

1. Assign meaningful labels to the document groups.
2. Create document groups that are responsive to the content of the documents in the search results.
3. Create document groups that correspond to the user's query.
4. Place documents in all appropriate groups.

In Chapter 5, I describe an evaluation that substantiates these claims.

However, the added functionality of dynamic categorization comes at a price. Unlike the other approaches, dynamic categorization can be applied only when the user's query matches one of the query types in the query model. Both classification and dynamic categorization also require some form of domain-specific knowledge. In classification, that knowledge takes the form of training sets. Dynamic categorization requires knowledge of the types of words and phrases used in that domain, and knowledge about the types of queries users make. In contrast, clustering or relevance ranking can be applied to any domain without representing any knowledge of that domain.

Chapter 3

System Specification

In Section 1.4, I specified four desirable characteristics for a document-organization system:

1. Create document groups that correspond to the user's query.
2. Assign meaningful labels to the document groups.
3. Create document groups that are responsive to the content of the documents in the search results.
4. Place documents in all appropriate groups.

In this chapter, I specify the system components necessary for obtaining those characteristics: the query model (Section 3.1), the terminology model (Section 3.2), the categorizer (Section 3.3), the organizer (Section 3.4), and the results-presentation interface (Section 3.5). Figure 3.1 shows how these components interact.

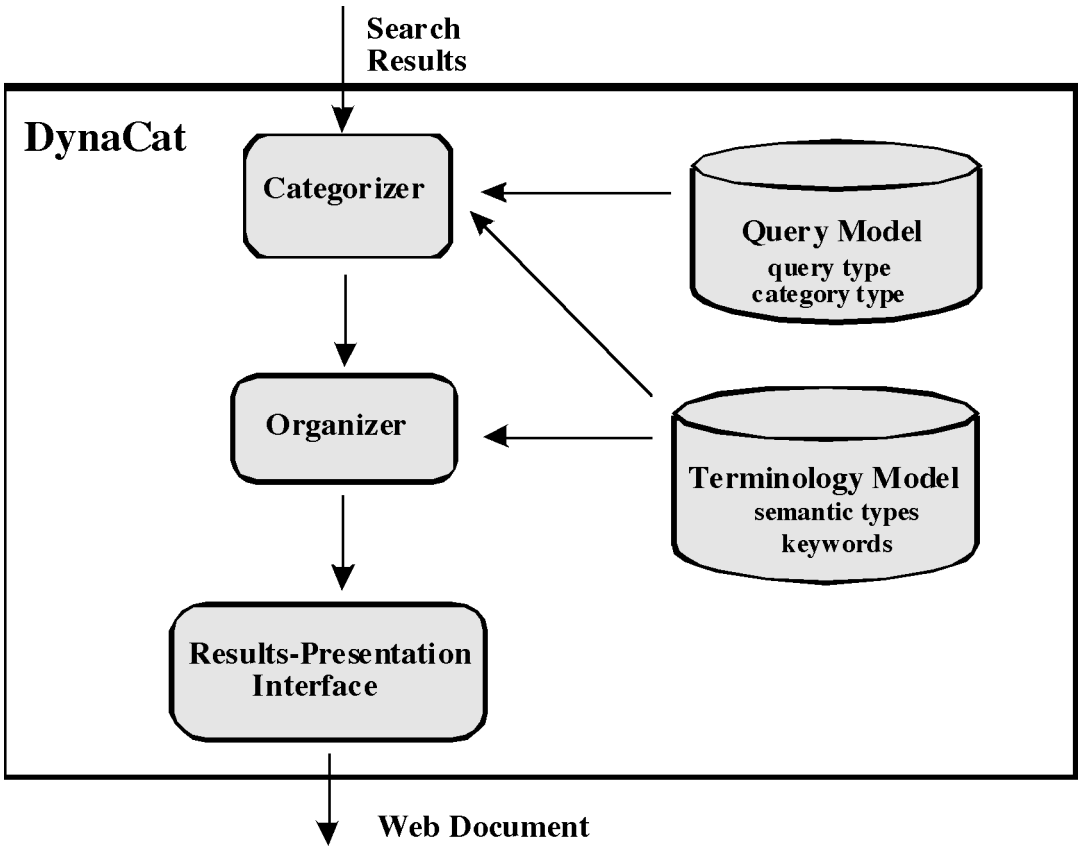


Figure 3.1 — DynaCat's system architecture. The categorizer takes the search results, and uses information from the query model and terminology model to produce a list of categories and documents assigned to those categories. The organizer takes the list of categories, and uses the hierarchical relationships among terms in the terminology model to create a hierarchical organization of those categories. The results-presentation interface takes the hierarchy of categories and converts it into an html file that can be displayed in any web browser.

3.1 Query Model

To organize the documents into categories that correspond to the user's query, the system needs knowledge about what kinds of queries users make in that domain, and about how search results from those queries should be categorized. The query model provides this information through the query types (Section 3.1.1), and category types (Section 3.1.2).

3.1.1 Query Types

It would be impossible to generate a comprehensive list of all the questions that people may want to ask, even if the question topics were limited to a specific domain such as medicine. However, it is possible to create an abstraction of the typical kinds of queries that people make. I created such an abstraction, called **query types**, for the domain of medicine. Query types, such as *treatment—problems* or *problem—preventive-actions*, are generalizations of common, specific queries, such as *What are the complications of a mastectomy?* or *What actions can I take to prevent breast cancer?* respectively. Because the query types are abstractions and thus are independent of specific medical terms, a small number of query types can cover many specific questions that a user might ask. For example, both specific questions *What are the complications of a mastectomy?* and *What are the side effects of taking the drug Seldane?* have the same *treatment—problems* query type, even though the questions refer to different treatments (e.g., the surgical procedure, *mastectomy*, and the drug *Seldane*).

For DynaCat's patient-oriented medical query model (see Table 3.1), I created nine query types that correspond to questions that patients ask when they look for information in medical journal articles. These query types may not provide comprehensive coverage of all questions that patients have, but the query types do cover many possible queries. For example, there are over 30,000 concepts in the medical terminology model that could be considered *problems*.[1] Since the query model contains seven problem-oriented query types, the model covers at least 210,000 specific, problem-oriented queries.

1. The number of problems was counted by adding together the total number of descendents of the terms *disease or syndrome*, *mental or behavioral dysfunction*, and *sign or symptom*, using the 1997 version of the UMLS.

Table 3.1. Patient-oriented medical query types and their typical forms.

Query Type	Form of Question
Prevention	
problem—preventive-actions	What can be done to **prevent** <*problem*>?
problem—risk-factors	What are the **risk factors** for <*problem*>?
Diagnosis	
problem—tests	What are the diagnostic **tests** for <*problem*>?
problem—symptoms	What are the warning signs and **symptoms** for <*problem*>?
symptoms—diagnoses	What are the possible **diagnoses** for <*symptoms*>?
Treatment	
problem—treatments	What are the **treatments** for <*problem*>?
treatment—problems	What are the **problems** that could result from <*treatment*>?
Prognosis	
problem—prognostic-indicators	What are the **indicators** that influence the *prognosis* for <*problem*>?
problem—prognoses	What is the **prognosis** for <*problem*>?

3.1.2 Category Types

For each query type, the system also needs an abstraction for the topics or categories that are appropriate for groups of search results. I call this abstraction **category types**. For example, when the user asks about the adverse effects of some drug, the types of categories that make sense are those that indicate the various adverse effects or *problems* that can arise when a person takes that drug.

The medical query model for DynaCat contains nine category types: *problems*, *symptoms*, *preventive–actions*, *risk–factors*, *diagnoses*, *tests*, *treatments*, *prognoses*, *prognostic–indicators*. As indicated by these names, each query type in the query model is linked to a cat-

egory type, which determines the kinds of categories that DynaCat will select whenever the user issues a query of that type.

By representing the category types separately from the query types, the system can link the multiple query types to the same category type, although currently the mapping is one-to-one. More importantly, this representation decision allows the system to provide a categorization option for queries that do not fit one of the predefined query types. Users could issue a normal search (without specifying a query type), and choose one of the category types as the way to categorize their search results. This separation also could enable the system to categorize the documents in a more interactive environment, where the user could select a subset of the search results and recategorize them according to a selected category type. The current system allows only the option to categorize documents by entering an initial query type.

3.1.3 Creation of the Query Model

To create a query model for a given domain, an implementor should think about the targeted user group, the kinds of questions that those users typically ask, and the kinds of information that is available to answer those questions. Ideally, the system implementor should analyze a set of questions that are frequently asked of that information source. First, she needs to identify all questions that have a list of possible answers and are likely to generate many relevant documents. Those questions are the ones that are appropriate for dynamic categorization. The implementor should look for ways to generalize from those questions to query types that have similar forms, and that have answers that she would group into similar types of categories. Her goal is to create a list of query types that cover most of the broad queries that users make when they are searching the document collection of interest.

In creating the medical query model for DynaCat, I thought about the kinds of medical questions that patients ask, and the kinds of information that is available in the medical literature. I also examined a list of frequently asked questions that at the Community Breast

Health Project (see Appendix B). Unfortunately, this list reflects all questions that breast-cancer patients asked, including those asked of physicians and other patients, rather than those made only when they were searching the medical literature. I discarded the questions that could not be answered from the medical literature. From these analyses, I generated the list of nine query types in Table 3.1. These nine query types cover at least two kinds of queries for each of the subject areas that most concerned the breast-cancer patients: diagnosis, treatment, prognosis, and prevention.

Other researchers have used similar abstractions of medical queries with clinicians as the targeted user group. The *clinical queries* component of PubMed provides canned MEDLINE queries that return information about diagnosis, prognosis, etiology, or therapy of a clinician-selected medical problem (NLM 1998b). Researchers from McMaster University created the search expressions that correspond to those clinical queries (Haynes, Wilczynski, et al. 1994). Researchers from Columbia University created a similar query abstraction called *generic queries* (Cimino, Aguirre, et al. 1993). Although none of these researchers have used their query abstractions to organize search results, their query abstractions are similar to those that I defined.

Although I created this query model for only medical patients, many of the query types generalize to any diagnostic domain. For example, if someone wanted to create a categorization system for documents on maintaining and repairing copy machines, queries such as *What should be done when the copies come out too light?* or *What problems could arise when someone adds new toner to the copier?* could map to the query types *problem—treatments* and *treatment—problems*, respectively. For such a domain, the system designer should be able to reuse these query types. However, she will need to use a different terminology model, and connect the existing category types to the appropriate concepts in the different terminology model. I explain the terminology model and this process in the following section (Section 3.2).

3.2 Terminology Model

To determine appropriate category labels for the document groups, the system needs to know which category labels are valid for the given category type. The **terminology model** provides this information by connecting individual **terms** (i.e., single words, abbreviations, acronyms, or multi-word phrases) to their corresponding general concept, called a **semantic type**. Those individual terms may become category labels if their semantic type is connected to the desired category type (see Section 3.3). For example, terms such as *AIDS*, *depression*, or *headache* could be category labels when the search results are organized by the category type *problems*, because their semantic types (*disease or syndrome*, *mental or behavioral dysfunction*, *sign or symptom*) correspond to the category type *problems*. This connection between the terminology model and the query model is illustrated in Figure 3.2. The system designer needs to only make the connections between the category types and the semantic types; she does not make any connections between the category types and specific category labels. This layer of separation allows maintenance of the specific terms (category labels) in terminal model independent from the maintenance of the query model. For example, if a new drug is discovered, the maintainers of the terminology model add the new drug to their model and connect it to the semantic type *pharmacologic substance*. The query model does change; yet DynaCat will be able to create categories labeled with the new drug name because the connections between the query model and the terminology model are made at an abstract level—between the category types and the semantic types.

3.2.1 Medical Terminology Model

DynaCat uses the medical-terminology models in the Unified Medical Language System (**UMLS**) (McCray, et al. 1993; Humphreys, Lindberg, et al. 1998). The National Library of Medicine maintains the UMLS, which contains four knowledge sources: the Metathesaurus, the Semantic Network, the SPECIALIST Lexicon, and the Information Sources Map. DynaCat uses the first two knowledge sources. The **Metathesaurus** contains information from more than 40 different medical vocabularies on over 476,322 unique con-

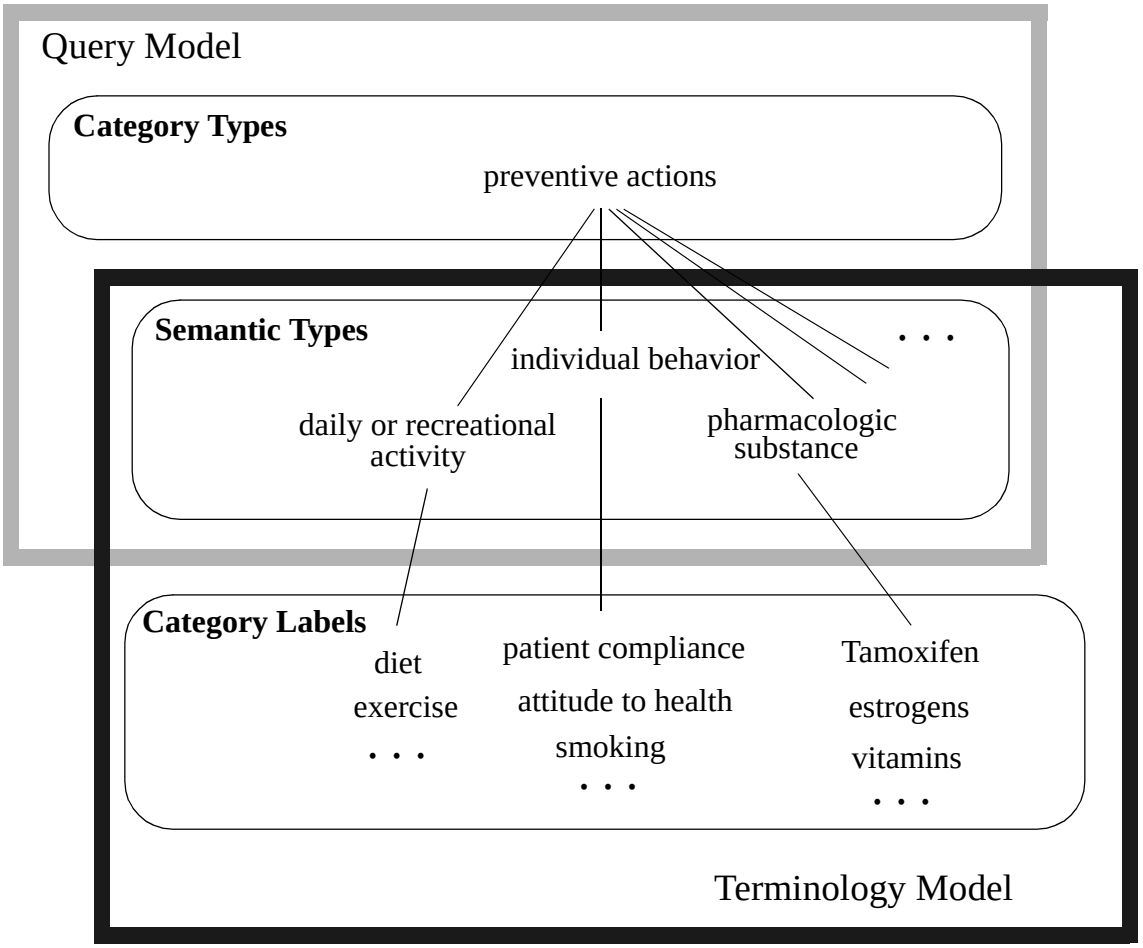


Figure 3.2 — The connection between the terminology model and the query model. This figure shows how the semantic types provide the link between the query model's category types and the terminology model's specific terms that are used to generate category labels. The semantic types are part of the terminology model; the system designer must explicitly link these semantic types to the category types of the query model. For example, when DynaCat organizes the search results according to the category type of *preventive actions*, it may create document groups with labels such as *diet*, *smoking*, or *vitamins*, because those terms have a semantic type that is linked to the *preventive actions* category type. The links between the semantic types and the category labels are part of the terminology. If the categorization system uses an existing terminology model, as DynaCat did, the system designer does not make those links; they were already made by the terminology model developers.

cepts named by more than 1,051,903 different biomedical terms (NLM 1999b). The Metathesaurus provides synonymy mappings among terms (even across multiple vocabularies), as well as *is-a* links between each term in the Metathesaurus and a term in the Semantic Network.

The **Semantic Network** represents semantic types and the relationships that can hold among them. The network contains 135 unique semantic types with major groupings for organisms, anatomical structures, biologic function, chemicals, events, physical objects, and concepts or ideas (NLM 1999c). These semantic types have been organized into an *is-a* hierarchy. All terms in the Metathesaurus contain links to their most specific semantic types.

3.2.2 Terminology Model Requirements

A terminology model that is used for dynamic categorization must represent most of the commonly occurring terms of the domain, including multi-word phrases. The model must provide links among those terms and their synonyms, abbreviations, or acronyms. Because only terms in the terminology model can become category labels, the comprehensiveness of the terminology model affects the accuracy of the categorization strongly. For example, if the terminology model does not contain the term *aspirin*, the categorizer will be unable to create a category with such a label when it creates categories of drugs.

The model also must provide some form of semantic link between specific terms and more general terms. The minimum requirement is for an *is-a* link between terms and high-level concepts that can be related to the category types appropriate for the domain. For example, the specific term *aspirin* needs to be connected through an *is-a* link to the more general term *drug*. Extensive links among terms, such as part-of links or causal links, may help systems create more accurate document categorizations than simple *is-a* links. For example, the knowledge that the drug *aspirin* treats the symptom *headache* would immediately inform the categorizer that *aspirin* would be a good category label when the user is asking about treatments for headaches. However, such correlations often are controversial, and such knowledge evolves rapidly. Few such detailed models exist, and even fewer are updated regularly. Therefore, dynamic categorization relies solely on the straightforward *is-a* hierarchy of terms.

3.2.2.1 Terminology Models for Other Domains

I have implemented dynamic categorization exclusively for the domain of medicine; however, the approach should be extensible to other domains that have large terminology models. Many terminology models for other domains exist and may be useful for categorizing documents (Rowley 1996). I describe a few of the popular models in the following paragraphs.

For computer science, the **Association for Computing Machinery** (**ACM**) created a taxonomy of computing terms (ACM 1997). The tree consists of 11 terms as first-level nodes, and each node has between five and 10 children. The total depth of the tree is four levels. An example path from the top level of the tree to the bottom level is *hardware—integrated circuits—types and design styles—gate arrays.* The ACM uses the terms in its model to label all articles published in its journals.

Mathematical Review sponsors a similar taxonomy of mathematics terminology (Review 1997). Its hierarchy has a depth of only three, but it is much broader than the ACM's, with more than 90 categories at the top level. The Review's coding system allows for up to 26 subcategories for each concept, each of which can have up to 99 subcategories.

Two general-purpose knowledge bases could be useful for categorization: WordNet (Fellbaum 1998) and CYC (Guha and Lenat 1994; Lenat 1995). WordNet is a database of over 118,000 English nouns, verbs, adjectives, and adverbs organized into synonym sets, each representing one underlying lexical concept. People can use **WordNet** online or they can download it (Miller 1997). **CYC** is a knowledge base of detailed, common-sense knowledge about more than 100,000 concepts. Although CYC is a proprietary knowledge base, Cycorp, the company that owns CYC, has released for public use about 3000 concepts from CYC's upper-level ontology (Cycorp 1997).

3.3 Categorizer

To determine which category labels are appropriate for the search results and the user's query, DynaCat needs the categorizer. The **categorizer** examines each document in the search results, determines what topics are discussed in that document, selects as category labels only those topics that match the desired category type, and assigns the document to its appropriate categories. The categorizer could use any of a variety of methods for determining what topics are discussed in a document. In Section 3.3.1, I discuss DynaCat's current approach to categorizing documents, and in Section 3.3.3, I present two other approaches that I explored but rejected.

3.3.1 Current Approach: Keyword Pruning

Many published documents contain keywords that authors or indexers have selected to describe the documents' content. The **keyword-pruning approach** takes advantage of this information in determining how to categorize search results. In the following sections, I describe the requirements for the terminology-model (Section 3.3.1.1), the keyword-pruning algorithm (Section 3.3.1.2), and an example of using this approach (Section 3.3.1.3).

3.3.1.1 Implications for Terminology-Model Requirements

This keyword-pruning approach requires that the documents in the collection have preassigned keywords, and that those keywords be represented in the terminology model. For the medical domain, DynaCat makes extensive use of one of the Metathesaurus's vocabularies, the Medical Subject Headings (MeSH). **MeSH** is a vocabulary of nearly 19,000 medical keywords that are organized into a hierarchy based on *is-a* relationships (NLM 1999a). For example, the MeSH term *penicillin* has the term *antibiotics* as a parent, which has the term *anti-infective agents* as a parent. The hierarchy is not a strict tree, in that terms may appear in multiple places in the hierarchy. For example, the MeSH term *pneumonia* has two parents: *lung diseases* and *respiratory tract infections*. Medical librarians, called MEDLINE indexers, manually assign the MeSH terms to documents in MEDLINE.

They are instructed to use the most specific MeSH terms that describe the content of the document (Humphrey 1992). They typically assign seven to 12 terms per document.

When the MEDLINE indexers assign a MeSH term to a document, they may further characterize the MeSH term by adding a subheading or qualifier. **MeSH subheadings** provide more information about how the term is used in the document. For example, if indexers assign the MeSH term *arthritis* with a subheading *etiology* to a document, this assignment indicates that the document contains information about the cause (etiology) of arthritis; this subheading might indicate, for example, that the article discusses arthritis as an adverse effect of a mastectomy. The system uses this added information in the form of constraints to improve its ability to categorize accurately for some of the query types. These constraints and the category type make up a query type's **categorization criteria**.

To improve the system's ability to include all appropriate categories, I added a possible criterion called **standalone subheadings**. It specifies a list of MeSH subheadings that indicate a category label should be created from any keyword that one of standalone subheadings modifies, even if that keyword's semantic type does not match those provided for the category type. Consider a case where the desired category type is *treatments*, and the standalone subheadings is *therapeutic use*. DynaCat would create a category label for any keyword that has the subheading *therapeutic use* assigned to it, because that keyword is likely to be a treatment regardless of that keyword's semantic type.

To improve the system's ability to discard inappropriate categories, I added another possible constraint called **required subheadings**, which indicates that a keyword must be modified by one of the required subheadings, in addition to having a semantic type indicated by the desired category type. As an example, the query type *treatment—problems* has a category type of *problems* and required subheadings of *etiology* or *chemically induced*. If the term *skin cancer* with the subheading *etiology* was assigned to a document, it would be selected as a category label because its semantic type is appropriate for *problems* and because it was modified by one of the required subheadings.

The overall effectiveness of the keyword-pruning approach is limited by the expressiveness of the terminology model and the accuracy of the keyword-assignment process. However, one large advantage of this approach is the ease of constructing the categorization criteria for each query type; after I created the first couple of categorization criteria, I could create new criteria for another query type within a few hours.

3.3.1.2 The Keyword-Pruning Algorithm

Because many of a document's keywords do not correspond to the user's query (see examples in Section 1.4.3), the categorizer must prune the irrelevant keywords from the list of potential categories. To accomplish this task, the categorizer examines each document in the set of results individually (see Figure 3.3). For each document, the categorizer examines each keyword. It looks up the keyword's semantic type in the terminology model, and compares that type to the list of acceptable semantic types from the categorization criteria. It also compares the keyword's subheadings to the required subheadings and the standalone subheadings in the categorization criteria. If a keyword satisfies all the categorization criteria, the categorizer adds the document to the category labeled with that keyword. If such a category has not already been created, it creates a new category labelled by that keyword. Every keyword in a document is checked against the categorization criteria; thus, each document may be categorized under as many labels as is appropriate for the given query type.

3.3.1.3 Example of the Categorization Process

To see how the keyword-pruning approach works, consider the example query: *What are the complications of a mastectomy?* and its corresponding categorization criteria in Table 3.3. Using this approach on the document in Figure 3.4, the system produces a categorization in which this document appears under only one category: *infections arthritis.* As is illustrated in Table 3.2, the document is not categorized under *diagnostic imaging, mastectomy,* or *middle age*, because those terms do not satisfy the category type constraint for the query type *treatment—problems*. The system does not categorize this document

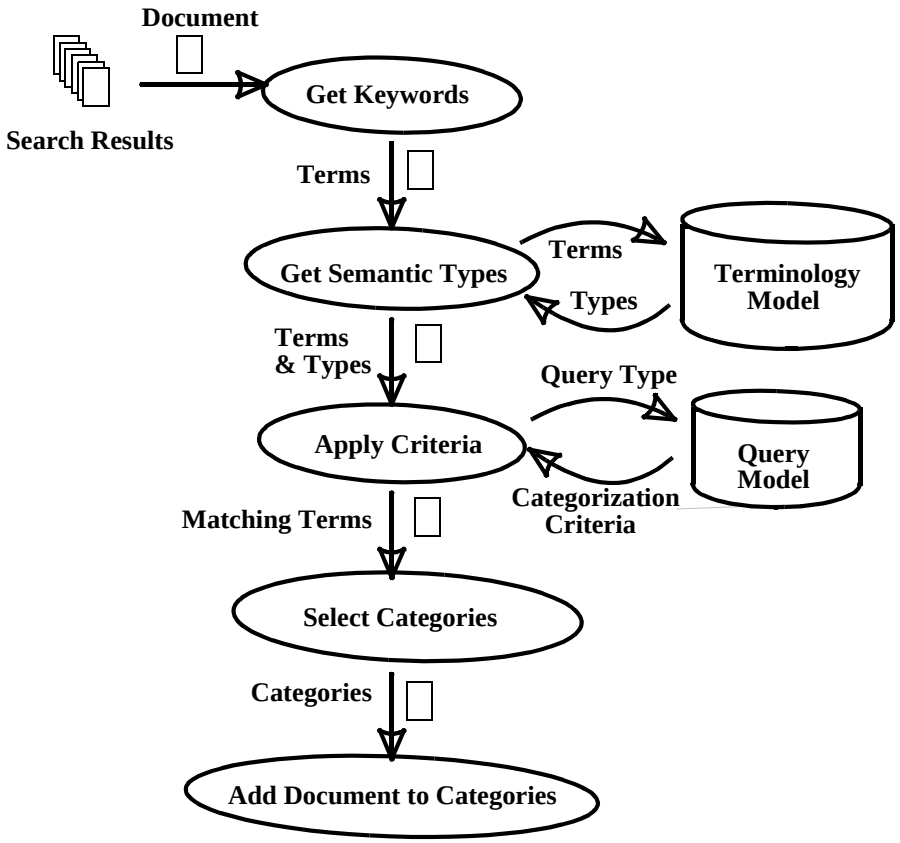


Figure 3.3 — Flow diagram for keyword pruning. For each document in the search results, the system gets all of that document's keywords. It retrieves each keyword's semantic types from the terminology model. If a keyword satisfies the categorization criteria from the query model, the system uses that keyword as a category label, and adds the document to the category.

under *lymphedema,* because the required subheadings constraint is not met.

3.3.2 tExploratory Approaches

I explored two other approaches to categorizing documents that I ultimately rejected as impractical. Both approaches were rejected before they were used in any evaluation of DynaCat. In Section 3.3.3.1, I explain the title-term spotting categorization approach and discuss its limitations. In Section 3.3.3.2, I describe the information-extraction categorization approach and the problems in scaling this approach.

Title: Septic Arthritis of the Shoulder After Mastectomy and Radiotherapy for Breast Carcinoma.
Author: Chaudhuri K, Lonergan D, Portek I, McGuigan L
Source: Bone Joint Surg Br; 75(2):318-21 1993
Type: JOURNAL ARTICLE
Language: ENG
Keywords: Aged, *Arthritis; Infectious -- diagnosis -- *etiology, *Breast Neoplasms -- *therapy, Combined Modality Therapy, Diagnostic Imaging -- etiology, Lymphedema, *Mastectomy -- *adverse effects -- methods, Middle Age, *Radiotherapy -- *adverse effects, *Shoulder Joint
Abstract: We report five patients who developed septic arthritis of the shoulder after cancer of the ipsilateral breast had been treated by surgery and radiotherapy. Lymphoedema was present in all cases. The infections were not obvious, having subacute onsets, and delays in diagnosis led to destruction of the joint in all but one patient.

Figure 3.4 — Example citation returned from a search on *mastectomy* and *adverse effects*. The capitalized terms in the list of keywords are the MeSH terms. If a MeSH term has a subheading, it appears in all lowercase letters, and it separated from the MeSH term that it qualifies by a double dash (--). The asterisk (*) indicates the main headings, which are the terms that the indexers thought were the main topics of the article.

Title: Septic Arthritis of the Shoulder After Mastectomy and Radiotherapy for Breast Carcinoma.
Author: Chaudhuri K, Lonergan D, Portek I, McGuigan L
Source: Bone Joint Surg Br; 75(2):318-21 1993
Type: JOURNAL ARTICLE
Language: ENG
Keywords: Aged, *Arthritis; Infectious -- diagnosis -- *etiology, *Breast Neoplasms -- *therapy, Combined Modality Therapy, Diagnostic Imaging -- etiology, Lymphedema, *Mastectomy -- *adverse effects -- methods, Middle Age, *Radiotherapy -- *adverse effects, *Shoulder Joint
Abstract: We report five patients who developed septic arthritis of the shoulder after cancer of the ipsilateral breast had been treated by surgery and radiotherapy. Lymphoedema was present in all cases. The infections were not obvious, having subacute onsets, and delays in diagnosis led to destruction of the joint in all but one patient.

Figure 3.5 — Example citation returned from a search on *mastectomy* and *adverse effects*. The capitalized terms in the list of keywords are the MeSH terms. If a MeSH term has a subheading, it appears in all lowercase letters, and it separated from the MeSH term that it qualifies by a double dash (--). The asterisk (*) indicates the main headings, which are the terms that the indexers thought were the main topics of the article.

3.3.2.1 Title-Term Spotting

The objective of the **title-term spotting** approach is to identify terms in a document's title that indicate that the document belongs to a category of interest. Many scientific articles,

Table 3.1. Categorization criteria for the query type *treatment—problems*.

Category type	Semantic types	Required subheadings
problems	disease or syndrome mental or behavioral dysfunction sign or symptom neoplastic process injury or poisoning sign or symptom	etiology

Table 3.2. Example keywords corresponding to the citation shown in Figure 3.4. The keyword in bold would become the category label for this citation.

Keywords	Subheadings	Semantic Types	Categorization Criteria Met
aged		age group	none
infectious arthritis	diagnosis, etiology	disease or syndrome	category type and subheading
breast neoplasms	therapy	neoplastic process	none
combined modality therapy		therapeutic or preventive procedure	none
diagnostic imaging	etiology	diagnostic procedure	subheading only
lymphedema		pathologic function	category type only
mastectomy		therapeutic or preventive procedure	none
radiotherapy		therapeutic or preventive procedure	none
shoulder joint		body space or junction	none

3.3.3 Exploratory Approaches

I explored two other approaches to categorizing documents that I ultimately rejected as impractical. Both approaches were rejected before they were used in any evaluation of DynaCat. In Section 3.3.3.1, I explain the title-term spotting categorization approach and discuss its limitations. In Section 3.3.3.2, I describe the information-extraction categorization approach and the problems in scaling this approach.

3.3.3.1 Title-Term Spotting

The objective of the **title-term spotting** approach is to identify terms in a document's title that indicate that the document belongs to a category of interest. Many scientific articles, have titles that summarize the content of the entire article; descriptive titles such as *Studies of a low-fat diet to prevent breast cancer* are common for medical journal articles. The title-term spotting approach takes advantage of this situation. It assumes that each document's title describes the content of the document accurately, and that the terms mentioned in the title reflect key concepts in the document. In the scientific literature, such as in MEDLINE, this assumption is reasonable; however, it is not a valid assumption for informal information, such as web documents.

Figure 3.6 illustrates the title-term-spotting algorithm. For each document in the search results, the system first identifies all the medical terms in the document's title. I use a term-identification tool developed at Lexical Technology, Inc., for this step. The term identifier uses a stop-word approach to identify potential noun phrases (Nelson, et al. 1994). It then checks the terminology model to determine whether the phrase or word is a known medical term. For example, given the title *"Angiosarcoma of the Breast Following Segmental Mastectomy Complicated by Lymphedema"* the term identifier returns the terms: *angiosarcoma, breast, segmental mastectomy, lymphedema.* The term identifier recognizes multi-word terms, such as *segmental mastectomy,* as well as single-word terms. Because every term in the user's query is present in every document returned, the categorizer removes any title terms that are also a term in the user's query or a synonym of a query term. This process prevents DynaCat from creating categories such as *breast cancer,*

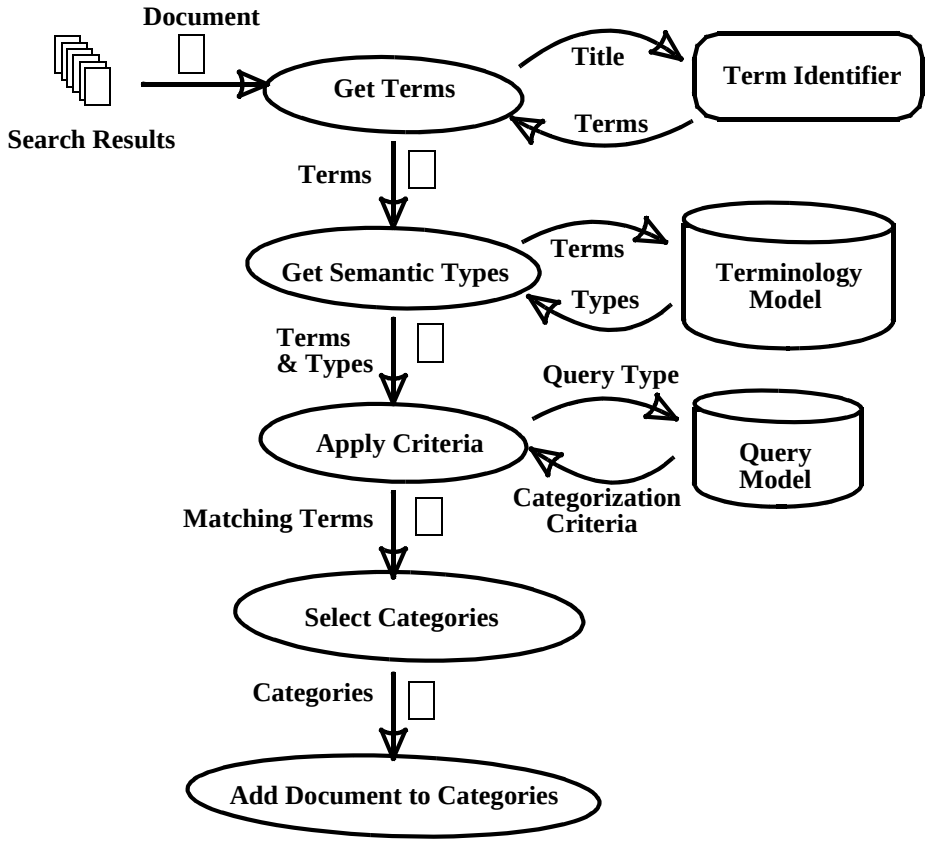


Figure 3.6 — Flow Diagram for title term spotting. For each document in the search results, the Term Identifier finds all of the medical terms in the document's title. Next the semantic types of each term are retrieved from the terminology model. For each term that satisfies the categorization criteria, the system uses the label selector function to determine the corresponding category label, selects the category with that label (or creates a new category with that label if one does not exist), and adds the document to that category.

which would contain every document in the search results, and thus would not be a useful category.

In the second step, the semantic types for each term are retrieved from the UMLS Semantic Network. Each term in the document is checked against the categorization criteria for the query type. If the term has a semantic type that corresponds to a category type in the categorization criteria, then the system retains that term as a category label. If no category already exists with that label, the system creates a new category and adds that document the category's document list. If a category already exists, then the system adds that document to the existing category's document list. This process is repeated for every term in

the document's title and then for every document in the search results. Since each term in the document's title is examined, the system can place a document in multiple categories.

As an example, consider the document with the title *"Angiosarcoma of the Breast Following Segmental Mastectomy Complicated by Lymphedema"* for the query *"What are the complications of a mastectomy?"* The query type that was chosen by the user is *treatment—problems* which lists *problems* as the category type. *Problems* corresponds to the following semantic types: *disease or syndrome, mental or behavioral dysfunction, pathologic function, neoplastic process, injury or poisoning, sign or symptom*. Table 3.3 shows each term in the document's title, that term's semantic type, and the categorization criteria that are satisfied. If a title term appears in boldface, then that title term satisfies applicable categorization criteria and is used as a category label for the document.

Table 3.3. Example of how the terms in the title "Angiosarcoma of the Breast Following Segmental Mastectomy Complicated by Lymphedema" could be used to categorize that document for a query on the complications of a mastectomy.

Title Term	Semantic Types	Categorization Criteria Met
angiosarcoma	neoplastic process	category type
breast	body part, organ, or organ component	none
segmental mastectomy	therapeutic or preventive procedure	none
lymphedema	pathologic function	category type

This document is added to two different categories: *angiosarcoma* and *lymphedema*. Notice that neither of those category labels is explicitly represented in the categorization criteria. The labels are selected automatically from terms used in the document. The document is not categorized under either *breast* or *segmental mastectomy* because those terms do not meet the categorization criteria.

There are two problems with this approach. First, this approach does not use any information about how the terms relate to one another, so it mistakenly assumes that any disease mentioned is a complication of the treatment. For example, the document title "*Acute Radiation Pneumonitis after Postmastectomy Irradiation: Effect of Fraction Size*" would be categorized under *pneumonitis*, even though pneumonitis is an adverse effect of radiation, rather than of a mastectomy. Second, the title may not reflect accurately the entire contents of the document. The document may discuss other complications that are not mentioned in the title, but this approach will not identify those complications. In contrast, the keyword-pruning approach would capture all discussed complications, as long as the indexer assigned the appropriate keywords to the document.

3.3.3.2 Information Extraction

Information extraction is a technique that identifies linguistic phrases and the relationships between the phrases to extract specific types of information from text. The goals in this approach to categorizing the documents are (1) to expand the repertoire of query types to include those whose categorization criteria do not map well to the simple presence of terms with specified semantic types, and (2) to improve the categorization accuracy.

Information extraction does not encompass in-depth natural-language understanding, instead it analyzes only portions of the text that match predefined templates. It extracts specific types of information, rather than analyzing the entire text to understand its content. Researchers have implemented many information extraction systems, mostly as part of the message understanding conferences (**MUC**) (MUC-3 1991, MUC-4 1992, MUC-5 1993, MUC-6 1995). These conferences were designed to foster research on large natural-language-processing systems for the automated analysis of military messages (Grishman and Sundheim 1996). For each conference, the organizers gave each participant a set of sample messages and instructions on the type of information to be extracted. The participants built their extraction systems based on this information, and shortly before the conference, the organizers gave them a test set which participants ran through their systems, without making any modifications to those systems. Each participant reported its results at the conference.

For this categorization approach, I used the systems implemented at the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts (CIIR 1997). These systems require both a dictionary that maps between specific terms and their parents, much like the relationship between terms and their semantic types in the UMLS, and a set of concept nodes. The **concept nodes** define the textual patterns that one wants to extract from the text. They specify either exact terms or parents, and the dictate how those terms should appear within the text. See Figure 3.7 for an example concept node. Researchers at the CIIR have created three tools for information extraction. The first tool, called MARMOT, is a text-bracketing tool. It segments text into sentences, assigns part-of-speech tags to words, and brackets text into annotated noun phrases, prepositional phrases, and verb phrases. The second tool, the BADGER extraction tool, uses the dictionary, the concept nodes, and the bracketed text to extract the desired text, as specified by the concept nodes. The third module, the CRYSTAL dictionary-induction tool (Soderland 1996, Soderland, et al. 1995), uses a set of training documents to automatically create text extraction rules that could be used by BADGER.

For each query type, the categorization criteria must contain a set of concept nodes, and a set of variables that get bound to the terms used in specific queries. For example, if the user's query is *"What are the factors that influence the prognosis for skin cancer?"*, the variable *<query-disease>* is bound to *skin cancer* and to that term's synonyms, descendants, and lexical variants from the Metathesaurus. See Figure 3.7 for an example concept node, modified for the categorizer approach.

Using the example titles from Figure 3.8 and the concept node in Figure 3.7, the information-extraction categorizer creates three categories: *clinical stage I* (which contains the documents titled *"Tumor Thickness and Prognosis in Clinical Stage I Malignant Melanoma"* and *"Prognosis of Clinical Stage I Melanoma Patients with Positive Elective Regional Node Dissection"*), *polypoidal* (which contains the document titled *"Prognosis for Polypoidal Melanoma Is Determined By Primary Tumor Thickness"*), and *advanced* (which contains the document titled *"Prognosis of Patients with Advanced Melanoma"*). Even though *skin cancer* (the user-specified disease query term) is not contained in any of

CN-type: disease-prognosis
Subtype: disease-modifier
Extract modifier from Prep.Phrase
Verb = <NULL>
Subject constraints:
    words include: “prognosis”
Prep.Phrase constraints:
    preposition = “of” or “for” or “in” or “with”
    head includes: <query-disease> or (synonym-of <query-disease>)
                or (descendant-of <query-disease>)
    modifier: <NULL>

Figure 3.7 — An example of concept node for the query type *disease—prognostic indicators*. This concept node matches to phrases such as *prognosis of stage I skin cancer* when *skin cancer* is the disease mentioned in the query.

“Age and Melanoma Prognosis”
“Tumor Thickness and Prognosis in Clinical Stage I Malignant Melanoma”
“Prognosis of Clinical Stage I Melanoma Patients with Positive Elective Regional Node Dissection”
“Prognosis for Polypoidal Melanoma Is Determined by Primary Tumor Thickness”
“Prognosis of Patients with Advanced Melanoma”

Figure 3.8 — Example titles from search on skin cancer prognosis. Documents such as these should be categorized under the factors affecting prognosis—such as *age*, *clinical stage*, or *tumor thickness*.

the example titles, the concept node matches to these titles because *melanoma* and *malignant melanoma* are descendants of *skin cancer* in the terminology model. Because no constraints are placed on the modifier for *melanoma* (as indicated by modifier: *<NULL>* in Figure 3.7), the system could create categorization labels from phrases that are not part of the terminology model.

The major problem with the information-extraction approach is the time and work required to create the concept nodes. Using the query type *problem—prognosis* as a prototypical example, I estimate that DynaCat would need at least 20 concept nodes to get reasonable coverage for a single query type. It took me several hours to create and test a single concept node; thus a system developer would need to devote over a week to create the concept nodes for one query type. Unfortunately, I could not use the CRYSTAL dictio-

nary-induction tool to help me create concept nodes. Normally, the developers of information-extraction systems want to extract information to answer specific questions; they are looking for precise patterns of explicit terms. DynaCat needs to extract information corresponding to query types and semantic types of terms, rather than specific questions and individual terms. CRYSTAL only learns concept nodes based on patterns of specific terms, and cannot learn based on patterns of semantic types of terms, which is what DynaCat requires. The information-extraction approach is the only option for categorizing informal documents, which do not have informative titles or keywords. However, to make this approach practical, the developer needs a tool to help her semi-automatically construct the concept nodes. It may be possible to create such a tool that is similar to CRYSTAL but can learn patterns of semantic types and query types. The development of such a tool was beyond the scope of my thesis, but I have explained how an information-extraction approach could be incorporated into DynaCat.

3.4 Organizer

When there are many relevant search results, often there are many relevant categories as well. For example, in the query about complications of mastectomy, the categorizer placed the 92 citations into 53 different categories. Because it is almost as overwhelming to deal with a list of 53 categories as it is to deal with a list of 92 citations, the system needs to organize the categories into an understandable hierarchy. The **organizer** creates this hierarchical organization of the categories. In the next sections, I present the additional domain-model requirements for generating such a hierarchy (Section 3.4.1), and the algorithm for the organizer (Section 3.4.2).

3.4.1 Additional Requirements for the Domain Models

In Section 3.2.2, I explained that the comprehensiveness of the terminology model is the most important factor in categorizing the documents. Whereas, in the organizer the depth of the terminology model is the critical factor. If the hierarchy has only two levels, such as

one level called *drugs* and another level for the specific drugs, then the system can create category labels for drugs such as *aspirin*, *ibuprofen*, or *penicillin*. But, if the system identifies 50 such drugs for category labels, the user may be as overwhelmed by the number of categories as she was by the number of documents. In such cases, a deeper term hierarchy helps the system group specific categories into a hierarchy with intermediate categories, such as *anti-inflammatories* or *antibiotics*. This hierarchy allows the user to get a quick summary of the kinds of categories present and allows her to pursue quickly only those topics that interest her.

DynaCat's organizer component also requires that the query model contain a field, called **starting parents**, that lists the top-level categories in the hierarchical organization. For most terminology models, the developer would not need to specify starting parents, because the concepts that she specified in the semantic types field would be the top-level terms in the categorization hierarchy. However, in the UMLS, DynaCat's medical terminology model, the semantic types are connected in a hierarchy to only other semantic types in the Semantic Network, rather than to the MeSH terms that DynaCat uses as category labels. This odd configuration of hierarchies is necessary because the UMLS contains many vocabularies that are not under the control of the NLM and therefore have their own term hierarchies. NLM uses the Semantic Network as a simple model to connect the many complicated vocabulary models to a small, manageable number of terms. However, because the semantic type hierarchy is separate from the hierarchy of MeSH terms that would be used as category labels, DynaCat cannot use the semantic types as the top-level categories in the categorization hierarchy.

3.4.2 Organizer Algorithm

The organizer could generate a hierarchy of categories by simply adding every category label's ancestors to a large tree, but this hierarchy would have two problems. First, many terms used as category labels have multiple senses, some of which may not be appropriate for the user's query. For example, the term *radiographic image enhancement*, can be considered a subfield of the physical sciences, but if a patient is asking about diagnostic tech-

niques for breast cancer, she cares only about the sense where it is diagnostic imaging technique. To solve this problem, the organizer uses only the category labels with an ancestor that is one of the starting parents for the given query type.

The second problem occurs because hierarchies that contain all of the category labels' ancestors are unwieldy and difficult to view. Many of the ancestors only clutter the tree; they provide little helpful information to the user. The organizer uses a maximum breadth threshold to help control the size of the categorization hierarchy and prune away unnecessary ancestors from the tree. For the studies described in Chapters 4 and 5, I set the maximum breadth threshold to ten. Decreasing the breadth of the hierarchy, however, often increases the depth of the hierarchy, which could also be undesirable. It may be appropriate to allow the user to set the maximum breadth threshold, so that she has control over this trade off between breadth and depth in an interactive environment.

The category organizer generates the hierarchy by taking following steps:

1. Merge synonymous categories.
2. Construct ancestor tree for all category labels.
3. Add starting parents as top nodes in categorization hierarchy.
4. For each node,
   - If total number of descendents that are document categories is greater than the maximum-breadth threshold, then add that node's direct children to the categorization hierarchy.
   - Otherwise, add all that node's document category descendents to the categorization hierarchy.
5. Repeat step 4, going down the ancestry tree until the maximum-breadth threshold is satisfied for each node or the bottom of the ancestry tree is reached.

In step 1, the system retrieves the synonyms for each category label from the UMLS. If two category labels are synonymous, they are merged into one category.

In step 2, the ancestors are retrieved for each category and added to the ancestor hierarchy for those search results. A category may have multiple parents, and therefore multiple ancestor paths. In such cases, all ancestor paths are added to the ancestor hierarchy. Note that if a category has multiple parents, that category could appear in multiple places in the final categorization hierarchy.

In step 3, the system generates the categorization hierarchy by first selecting the top nodes of the ancestry tree that match the starting parents from the query model. The system then adds these nodes to the top level of the categorization hierarchy. All the ancestor paths that are not derived from the starting parents are discarded. The resulting categorization hierarchy contains only the subset of the terminology model that is related to the category labels.

In step 4, the system prunes the categorization hierarchy using the preset maximum-breadth threshold. For each of the top-level nodes in the categorization hierarchy, the number of candidate document categories that are its descendents are counted. The number of descendents of a node includes all the document categories that have labels that are direct children or indirect children of that node's name. For example, in Figure 3.9, the node named *disease* has four candidate document categories as its descendents (*surgical wound infection, bacteremia, staphylococcal infections,* and *infectious arthritis*), even though none of them are its direct children. If the number of document-category descendents for a node is greater than the maximum-breadth threshold, then that node's direct children are added to the categorization hierarchy. If the threshold is not exceeded, then all that node's document-category descendents are added directly to the categorization hierarchy. This process is repeated until all the document categories have been added to the categorization hierarchy.

This process makes the final categorization hierarchy responsive to the distribution of documents from the search results. When there are many categories at one level, the categories are grouped under a more general label, when the term hierarchy can provide such a label. In cases where the no general label is present, the categorization hierarchy nothing can be done to reduce the breadth; thus, it will exceed the maximum-breadth threshold.

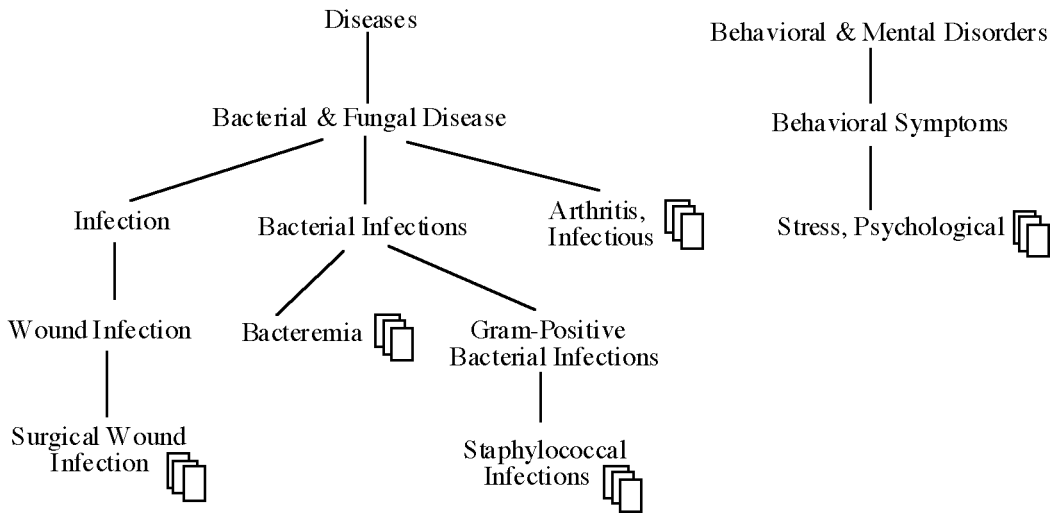


Figure 3.9 — Example of the MeSH ancestry tree for the original category labels (indicated by the document icon).

As an example, consider the case where there are five categories generated: *surgical wound infection, bacteremia, staphylococcal infections, infectious arthritis,* and *psychological stress*. The MeSH ancestry tree for those category labels is illustrated in Figure 3.9. If the maximum-breadth threshold is set to four, the organization in Figure 3.10 would result. If the threshold were set to three, the final categorization hierar-

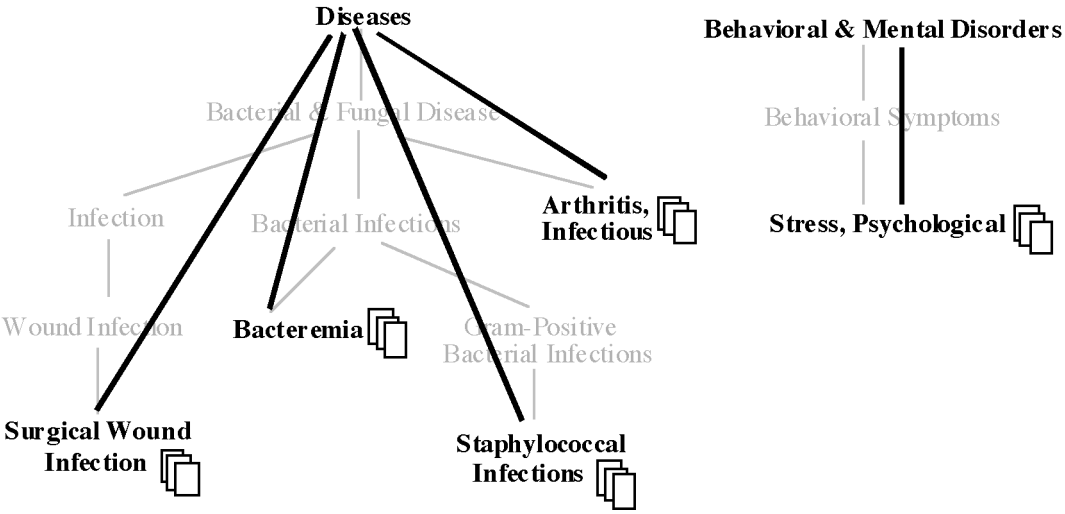


Figure 3.10 — Organization for a maximum breadth threshold of four.

chy would be deeper and less broad, as shown in Figure 3.11.

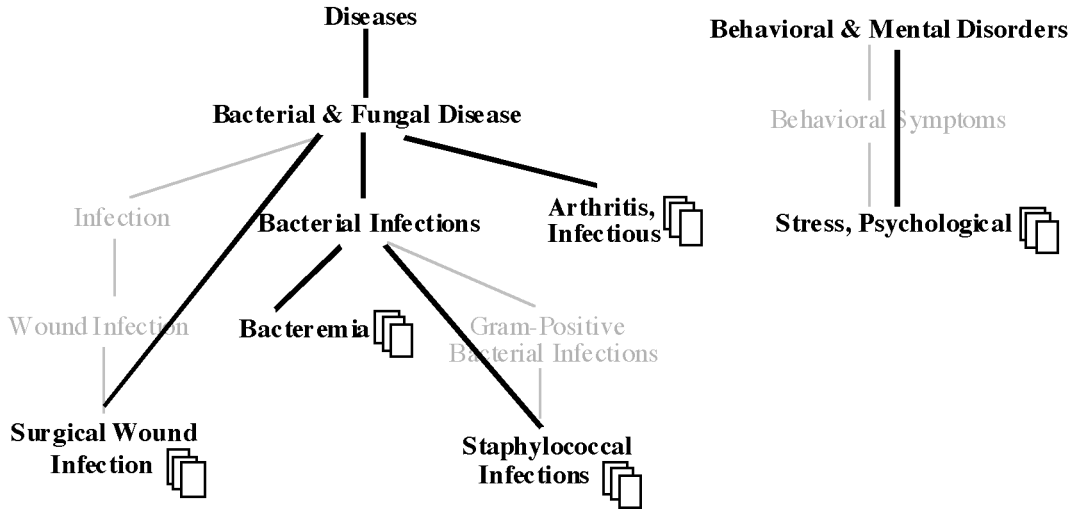


Figure 3.11 — Organization for a maximum-breadth threshold of three.

3.5 Results-Presentation Interface

The **results-presentation interface** takes the hierarchical organization of categories from the organizer and produces a web document. Figure 3.12 illustrates the web document that DynaCat generated for the search on adverse effects of a mastectomy. The document is split into three frames: one horizontal frame or row at the top of the document, and two vertical frames or columns at the bottom of the document. The top frame always contains the query and the number of different citations that satisfied the query. The left frame contains the most general categories; this frame is designed to be used like a table of contents for a book, such as in the design of the electronic SuperBook (Egan, et al. 1989). The numbers in parentheses indicate the number of unique citations or references in the named category and provide hyperlinks to the corresponding category as they appear in the entire categorization structure. The right frame can contain either the entire hierarchical organization of categories with the titles of the citations that belong to each category, or the entire citation. The citation's title in the categorization hierarchy is a hyperlink to the entire citation, including the document's title, author, source, type, language, unique identifier, subject headings, and abstract.

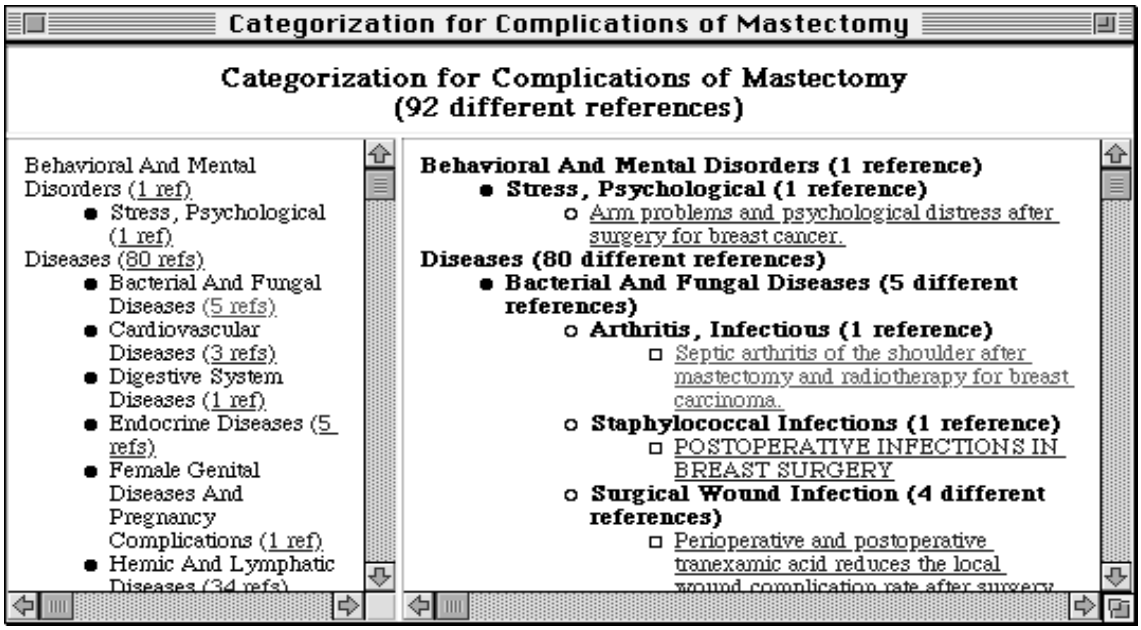


Figure 3.12 — DynaCat's interface. The interface is broken into three frames or window panes. The top window pane displays the user's query and the number of documents found. The left pane shows the categories in the first two levels of the hierarchy. This pane provides a table-of-contents view of the organization of search results. Clicking on the number in parentheses brings that section of the hierarchy to the top of the right pane. The right pane displays all the categories in the hierarchy and the titles of the documents that belong in those categories. Each document's title is a hyperlink to that document's citation. Clicking on the hyperlink causes the corresponding citation to appear in the right pane, replacing the list of all the categories and document titles. The complete list of categories and document titles will be displayed in the right pane again if the user clicks on any of the hyperlinks in the table-of-contents pane.

3.6 Summary

In this chapter, I described the components of dynamic categorization. I detailed the domain-specific knowledge that is in the form of two domain models: a terminology model, and a query model. I presented the system architecture, and specified each component. Note that the system that I described is only a research system, and is not available for general use. There was no user interface for specifying the user's query and query type other than specifying them as arguments to a LISP function.

In Chapter 4, I describe the user study that compares the usefulness of DynaCat to that of a document-clustering tool, and a relevance-ranking tool.

Chapter 4

Usefulness Evaluation

In Chapter 1, I stated my general hypotheses that dynamic categorization will organize the results of a search into a hierarchy of categories, and that this organization will help users to understand and explore their search results. In this chapter, I discuss the evaluation that tested one component of the hypothesis: whether the organization helps users. I present the objectives of this evaluation (Section 4.1), describe the systems that I compared to Dyna-Cat (Section 4.2), outline the pilot study (Section 4.3), and report the final study (Section 4.4).

4.1 Objectives

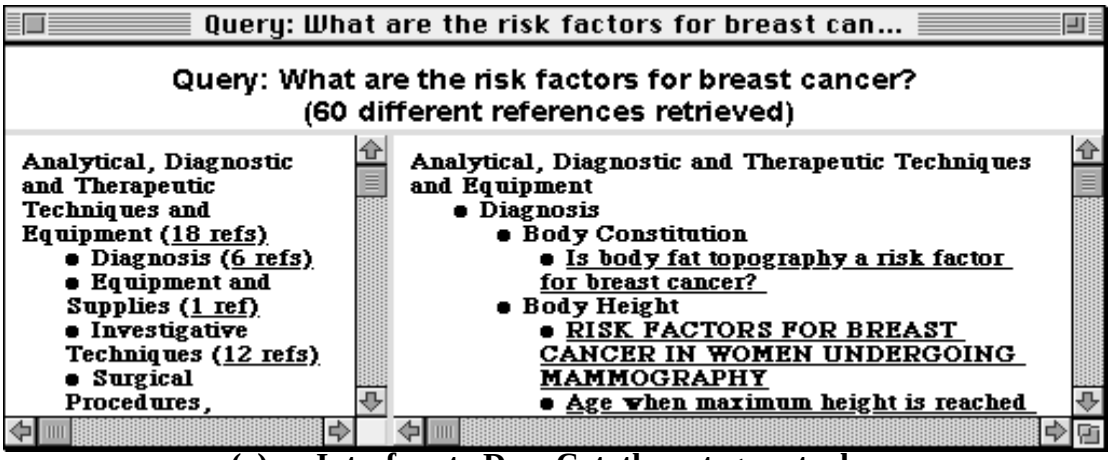
My general goal in this evaluation was to determine how useful the system is at helping users to understand and to explore their search results. Specifically, I tested the claim that organizing search results using dynamic categorization will be more useful to users who have general questions than are the two other approaches: relevance ranking and clustering. I define a useful system as one that helps users

- To learn about the kinds of information that pertain to their query
- To find answers to their question efficiently and easily
- To feel satisfied with their search experience

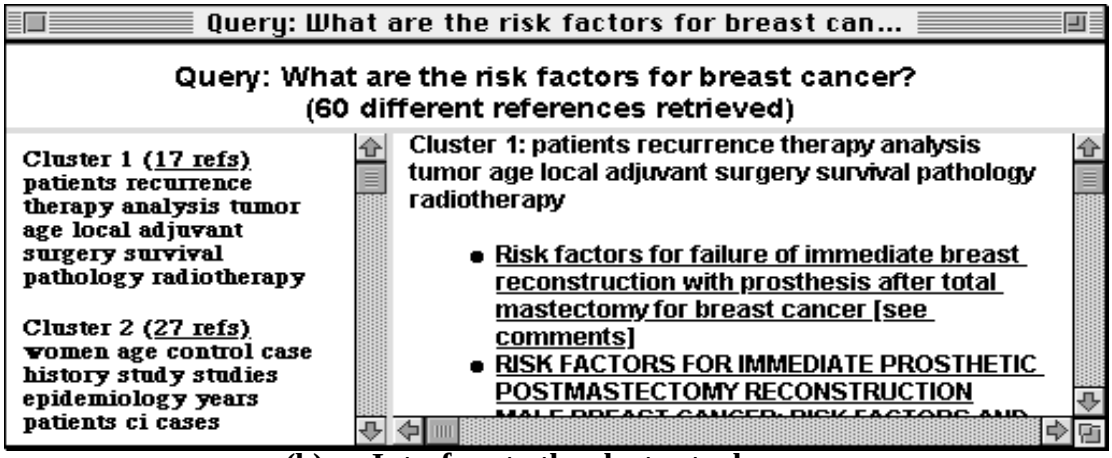
For this evaluation, satisfaction includes the subjects' perception of many attributes such as the clarity of the organization of search results, the ease of tool use, the usefulness of the organization, and the accuracy of the organization. See Appendix A for the complete satisfaction questionnaire.

4.2 Comparison Systems

In this evaluation, I compared DynaCat to two other systems that organize search results. Each subject used all three organizational tools: (1) DynaCat, (2) a tool that ranks the search results according to relevance criteria, and (3) a tool that clusters the search results. My intent was to measure the effect of the organization of the documents, rather than the effect of individual user interfaces. Therefore, I made the interfaces to the three tools as similar as possible. For example, the relevance-ranking tool divides the documents into groups of 10 based on the documents' relevance scores. The relevance tool displayed the relevance groups in the same way that the clustering tool displayed clusters and that DynaCat displayed its categories. Figure 4.1 shows example interfaces for each of the three tools.



(a) — Interface to DynaCat, the category tool.



(b) — Interface to the cluster tool.

(c) — Interface to the ranking tool.

Figure 4.1 — The interfaces to DynaCat (a), the cluster tool (b), and the ranking tool (c). All interfaces are divided into three frames or window panes. The top window pane displays the user's query and the number of documents found. The left pane provides a table-of-contents view of the organization of search results. The right pane displays all the document titles using the organization scheme of the tool.

I describe the tool for relevancy ranking in Section 4.2.1 and the tool for clustering in Section 4.2.2.

4.2.1 Relevance-Ranking Tool

Search systems have used many different algorithms for ranking search results (see Section 2.3), and researchers have studied the effectiveness of certain algorithms in different situations (Salton and Buckley 1988; Efthimiadis 1993). The relevance-ranking tool for this evaluation uses a standard algorithm recommended by Salton for situations in which the queries are short and the vocabulary is technical (Salton and Buckley 1988). This algorithm uses the following formulae to calculate a document's similarity or relevance score:

$$\text{new\_tf(i)} = 0.5 + 0.5 \times \frac{\text{tf(i)}}{\text{maxtf}}$$

$$\text{wt(i)} = \text{new\_tf(i)} \times \log\frac{\text{N}}{\text{n}}$$

$$\text{wq(i)} = \frac{0.5 + 0.5 \times \text{qf(i)}}{\text{maxtf}} \times \log\frac{\text{N}}{\text{n}}$$

$$\text{Relevance Score} = \frac{\sum_{i=1}^{T} (wq(i) \times wt(i))}{\sqrt{\sum_{i=1}^{T} wq(i)^2 \times \sum_{i=1}^{T} wt(i)^2}}$$

where N = total number of documents in the collection.

T = total number of terms in the collection.

n = the number of documents that contains the term.

tf(i) = the frequency of term *i* in the document.

qf(i) = the frequency of term *i* in the query.

maxtf = the maximum term frequency for any term in the document collection.

I implemented this ranking algorithm in Common LISP. Each word in the documents was considered a term, but none of the words were stemmed. I made the interface for the ranking results similar to the interface to that for DynaCat (see Figure 4.1).

4.2.2 Clustering Tool

I used the SONIA document-clustering tool as a comparison system (Sahami, Yusufali, et al. 1998). SONIA uses a two-step approach to clustering documents: it uses group-average hierarchical agglomerative clustering to form the initial set of clusters, then refines the clusters with an iterative method. I provided the search results for each query as a set of html documents, and Mehran Sahami, the creator of SONIA, sent me back a set of documents indicating the number of clusters created, the words that described each cluster, and the set of documents that SONIA assigned to each cluster. He used the default settings for SONIA, and had it find the maximum number of clusters. I wrote an interface to read his files and to present the results in an interface that is similar to that of DynaCat (see Figure 4.1).

4.3 Pilot Study

I conducted a pilot study to determine (1) whether any of the questions, instructions, or tasks were confusing; and (2) how many subjects would be needed for statistically significant results. I recruited five volunteers for the pilot study. All were women. Two were already familiar with my research; the other three had no previous exposure to DynaCat or to my research. The first subject made many suggestions for clarifying the instructions and the user-satisfaction questions. I used her feedback to revise the instructions and the questions. The next three subjects used those revised forms. I used the data from those three subjects in a power calculation to determine the appropriate number of subjects necessary to obtain significant results. Using the software from the Biostatistics Primer (Glantz 1997), I determined that I would need between 8 and 15 subjects to achieve significant results. I decided to try to recruit 15 subjects.

As a result of the pilot study, I made a few more changes to the wording of the user-satisfaction questions, and to the tutorials on each of the tools. I also modified the tasks slightly, as I describe in Section 4.4.1.2. The fifth pilot subject used the revised version and indicated that all instructions and questions were clear. None of the tools were modified during or after the pilot study.

4.4 Final Study

In this section, I describe the final study of usefulness. I explain the evaluation methods (Section 4.4.1), and report the study results (Section 4.4.2).

4.4.1 Methods

For this evaluation, I used methods from the field of human-computer interaction, unlike most evaluations of information-retrieval systems, which use precision and recall measures exclusively. Although no other study is exactly like the one that I designed, I was inspired to use some of the methods by the study designs of SuperBook (Egan, Remde, et

al. 1989) and Scatter/Gather (Pirolli, Schank, et al. 1996). In the following sections, I outline the methods for this evaluation. I describe the subjects of the study (Section 4.4.1.1) and the procedure that these subjects followed (Section 4.4.1.2).

4.4.1.1 Subjects

The subjects for this study were breast-cancer patients or their family members. I recruited these subjects via the Community Breast Health Project (CBHP 1997), the Stanford Health Library, and Stanford University's Oncology Day Care Center. Each subject signed a written consent form before participating; each was paid. A total of 17 subjects participated in the final study; however, I used the data from only 15 subjects. I did not include any of the data from the first two subjects in the final results because they did not use the 1-to-5 scale for answering many of the user-satisfaction questions, they misunderstood the directions for the timed tasks, and they neglected to answer a couple of other questions. I checked all the answers of the remaining 15 subjects before I allowed them to go on to the next segment of the evaluation; I thus, made sure that they understood the directions prior to doing each task. The subjects knew that the purpose of the study was to investigate the usefulness of three search tools: a category tool (DynaCat), a cluster tool, and a ranking tool. They did not know that I created one of the tools.

4.4.1.2 Procedure

Every subject used all three organizational tools: (1) the category tool (DynaCat), (2) the cluster tool (described in Section 4.2.2), and (3) the ranking tool (described in Section 4.2.1). Each subject used three different queries. I randomized the query used with each tool and the order in which the subjects used the tools. See for a graphical illustration of the procedure. Each subject followed this procedure:

1. Filled out a human subjects consent form.
2. Filled out a background questionnaire by answering the following questions:
   - Do you have breast cancer or have you ever had breast cancer?
   - Do you have a relative or close friend who has breast cancer?

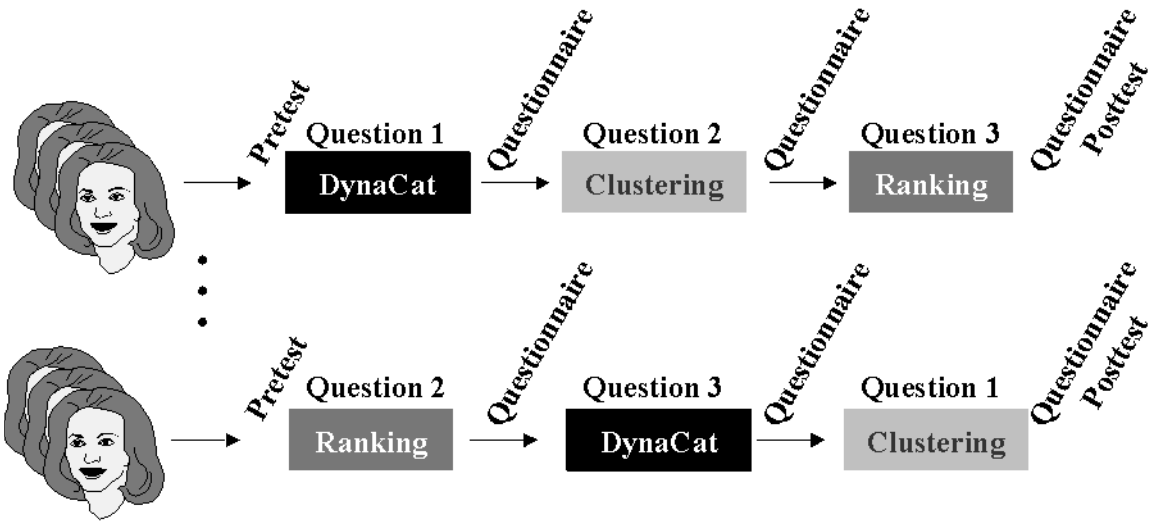


Figure 4.2 — The usefulness evaluation study design.

- Have you ever read anything about breast cancer?
- Have you ever searched for information about breast cancer?
  - In books?
  - In popular magazines?
  - In medical journals?
  - In MEDLINE?
  - On the web?

3. Answered the following questions on how much she or he knew about the subject of the queries:
   - List all of the treatments for breast cancer that you can think of:
   - List all of the ways to prevent breast cancer that you can think of:
   - List all of the factors that influence breast cancer prognosis that you can think of:

4. Read and followed the tutorial for each of the three tools: category tool (Appendix C), cluster tool (Appendix D), and ranking tool (Appendix E). Each tutorial used the query, *"What are the risk factors for breast cancer?"*
5. Given a tool and a query, completed three timed tasks to find specific information (Appendix F):
   - Find as many answers to the original query as possible in 4 minutes.
   - Find a document that answers a specific question related to the original query, and record the time that it took to find the answer.
   - Find a document that answers a different, specific question related to the original query, and record the time that it took to find the answer.
6. Filled out the user-satisfaction questionnaire for the tool that he or she just used (Appendix A).
7. Repeated steps 5 and 6 for the remaining two tools and queries.
8. Answered the original questions on how much she or he knew about the subject of the queries, not counting his or her original answers from step 3. (Subjects did know that there would be a posttest).
9. Answered the following questions:
   - Which tool did you like best? (Ranking Tool, Cluster Tool, or Category Tool) Why?
   - Which tool did you like least? (Ranking Tool, Cluster Tool, or Category Tool) Why?
   - Did any of the tools help you learn more about the topic of the question? If so, which one?

The order of the tutorial exposure was the same as the order of tool use. Because each subject used each of the tools before starting the measured part of the study, I assumed that the order of tool use would not influence the results.

DynaCat generated the search results by querying the CancerLit database through the Oncology Knowledge Authority (Tuttle, Sherertz, et al. 1994). It limited the results to

documents that were written in English and that contained an abstract. I chose three general queries that represented the kinds of questions that patients typically ask, and that were general enough to have multiple answers, and thus would be appropriate for an information-exploration tool, such as DynaCat. The three queries that I used were *What are the prognostic indicators for breast cancer?*, *What are the treatments for breast cancer?*, and *What are the preventive measures for breast cancer?*. I also provided the corresponding query types: *problem—prognostic-indicators*, *problem—treatments*, and *problem—preventive-actions*. I chose these three queries for the evaluation because the number of documents returned were similar (between 78 and 83 documents), and I did not want the number of documents returned to influence tool performance.

To create the specific questions for step 5, I asked an oncologist what he expected a patient to learn after reading documents returned from the different queries. In the pilot study, both of the timed questions came from him. However, for some of his questions, the subject could not determine which documents could answer the question by looking at the title of the documents, even when the abstract of the document contained the answer. In these cases, the subjects became extremely frustrated when they were using either the cluster tool or the ranking tool. They often gave up before they could find a document that was relevant to the question in their task. No subject experienced this difficulty using the category tool, because the category labels indicated when a document discussed the topic related to the question. Even though the results were better with the category tool, I decided to use only questions that related to topics that were visible in some document's title for the final study. I made this decision because the subjects became upset when they could not find an answer, and because I would have difficulty comparing the timed tasks if people gave up. In the final study, I chose one question from the oncologist and one from the list of frequently asked questions gathered from the Community Breast Health Project (Appendix B). For both questions, I chose the first question that was answered by one of the documents in the search results, that met the criterion of being visible in at least one document's title, and that had either a yes-or-no answer or a simple, one-word answer.

I made one other change to the timed questions from the pilot study: I asked the subjects from the pilot study to find a document relevant to answering the timed questions, rather than asking them to answer the questions, as I did in the final study. I noticed a large variation in which documents the subjects thought were relevant. One subject would pick simply the first document that mentioned the topic of the question; another subject would examine many documents that mentioned the subject before choosing the document that discussed the topic in the most detail. This discrepancy in the interpretation of relevance led to a large variation in the time that it took the subjects to complete the tasks. Thus, the timing data did not indicate which tool helped subjects find the answers most efficiently. I hoped to alleviate this problem by changing the task from finding a relevant document to finding any document that answered the question and stating the answer. Unfortunately, this formulation of the tasks created other problems, as described in Section 4.4.2.1.

I instructed the subjects to answer the timed questions as quickly as they could. I promised the subjects that I would let them use any of the tools after the study if they wanted to look for information to satisfy their own information needs. I allowed the subjects to use the tool again when they answered the user satisfaction questionnaire, but I cleared the screen before they answered the post-test questionnaire.

4.4.2 Results

In this section, I discuss the results from the final usefulness study. I discuss the results of the timed tasks (Section 4.4.2.1), the amount the subjects learned during the study (Section 4.4.2.2), their satisfaction with the search process (Section 4.4.2.3), and their answers to the open-ended questions and comments (Section 4.4.2.4).

4.4.2.1 Timed Tasks

All subjects completed two types of timed tasks. First, they found as many answers as possible to the general question (e.g., *What are the preventive actions for breast cancer?*) in 4 minutes. The subjects' second type of task was to find answers to two specific questions (e.g., *Can diet be used in the prevention of breast cancer?*) that related to the original,

general query. I combined the results of the second type of task into one mean value: the time to answer specific questions. See 4.1 for a summary of the results for the timed tasks.

Table 4.1. Results for the timed tasks.

	DynaCat	Cluster	Ranking	*p* value D vs C	*p* value D vs R
Answers found in 4 minutes	7.80	4.53	5.60	0.013	0.004
Time (minutes) to find answers to specific questions	2.15	2.95	2.21	0.274	0.448

To determine whether there was a significant difference among the three tools, I first used a repeated-measures analysis of variance (ANOVA). Because I was interested in only whether the category tool performs better than the cluster tool or better than the ranking tool, I also used a paired, one-tailed *t* test to determine the level of significance in comparing DynaCat (D) to the cluster tool (C), and in comparing DynaCat to the ranking tool (R). Using the repeated-measures ANOVA, I found a significant difference ($p = 0.035$) among the tools for the number of answers that the subjects found in 4 minutes. When the subjects used DynaCat, the category tool, they found nearly twice as many answers as they did with the other two tools. This difference was significant when I used the paired *t* test as well. Note that, although the mean number of answers found with the ranking tool was greater than that found for the cluster tool, the *p* value was lower in the comparison of DynaCat to the ranking tool than it was when comparing DynaCat to the cluster tool. This result occurred because the subjects consistently found fewer answers with the ranking tool than they did with DynaCat; whereas their results with the cluster tool were variable.

There was no significant difference across the tools for the time it took the subjects to find answers to specific questions. As in the pilot study, the time it took subjects to find documents that answered the specific questions varied greatly. In this final study, I noticed two sources of this variability. The first source was the position of a document containing an answer to the question within the relevance-ranked list. For one question, it was obvious from the title of the first document in the relevance-ranked list that it answered the ques-

tion, thus the time that a subject took to answer that question was very small if she used the ranking tool. Second, I observed that several subjects answered the question based on only the title of the document, whereas most other subjects read the entire abstract before answering the question. Reading the abstract took much longer than simply reading the title, particularly because the terminology in those abstracts was technical and sometimes was completely unfamiliar to the subjects. Thus, the time to read the abstract, rather than the time to find a document among the search results, most heavily influenced the time to find an answer. In future studies, I will collect and analyze the documents visited and the paths that users follow when they use each tool. Such information could provide futher insights into users' behavior for exploratory search tasks. Unfortunately, I did not collect such data in these experiments.

4.4.2.2 Amount Learned

To determine the amount that subjects learned during the study, I gave each subject a pre-test and a posttest of their knowledge related to the 3 breast-cancer questions (see steps 3 and 8 in Section 4.4.1.2). I measured the number of new answers on the posttest. The mean number of answers learned for the category tool (2.80) was greater than those for the cluster tool (2.20) and for the ranking tool (2.33); however, this difference was not statistically significant. The largest influence on this measurement was the order in which the subjects looked for answers to the question. Subjects remembered fewer answers from their first question (1.93) than they did from their second (2.80) or third (2.60). Using a paired, one-tailed *t* test, I found the difference between the times of the first and second questions to be significant ($p = 0.04$). However, the difference between the second and third questions was not significant ($p = 0.36$), possibly because the subjects could still remember answers from their second question, about 30 minutes in the past, but had more difficulty remembering answers from their first question, nearly an hour in the past. The tool used may have had an influence on the amount learned, but the number of answers that the subjects remembered for the posttest was correlated more strongly with how recently the subjects found answers to that question, rather than which tool they used.

4.4.2.3 User Satisfaction

To measure user satisfaction, I used both a validated satisfaction questionnaire (Dolland-Torkzadeh 1988), and a questionnaire that I created to measure other important types of satisfaction. Appendix A shows the combined questionnaire that I used. Subjects filled out the questionnaire for each of the three tools.

Questions 1 through 10 were from the validated questionnaire, although I modified questions 3, 5, and 10 slightly to match each tool more closely. Figures 4.3 and 4.4 show the results from the validated questionnaire. The subjects answered the questions using a scale

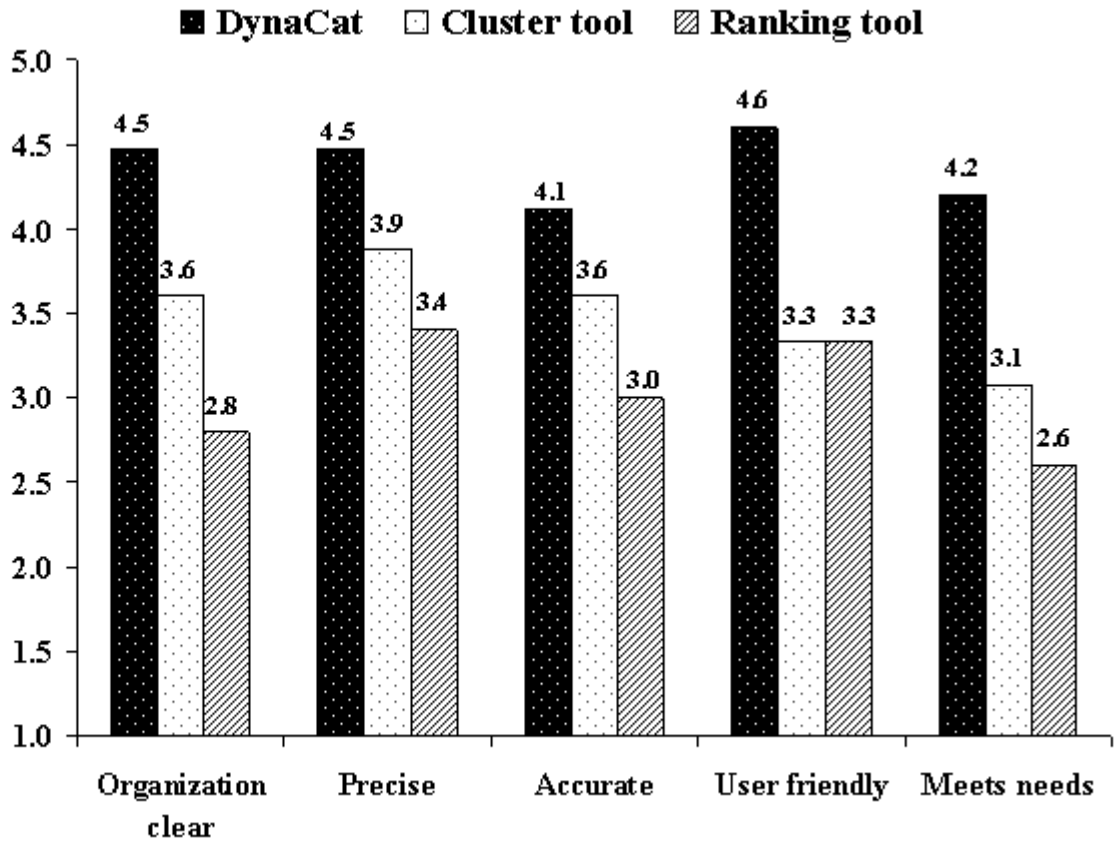


Figure 4.3 — Results from the first five questions from the validated user-satisfaction questionnaire. The mean values across all 15 subjects are shown on the *y* axis. The *x* axis shows a brief summary of the questions asked, numbered 1 through 5. The full questionnaire is given in Appendix A. Subjects answered the questions using a scale from 1 to 5, where 1 meant *almost never* and 5 meant *almost always* (the ideal answer). The difference between DynaCat and the cluster tool was statistically significant ($p < 0.05$) for all five questions, as was that between DynaCat and the ranking tool.

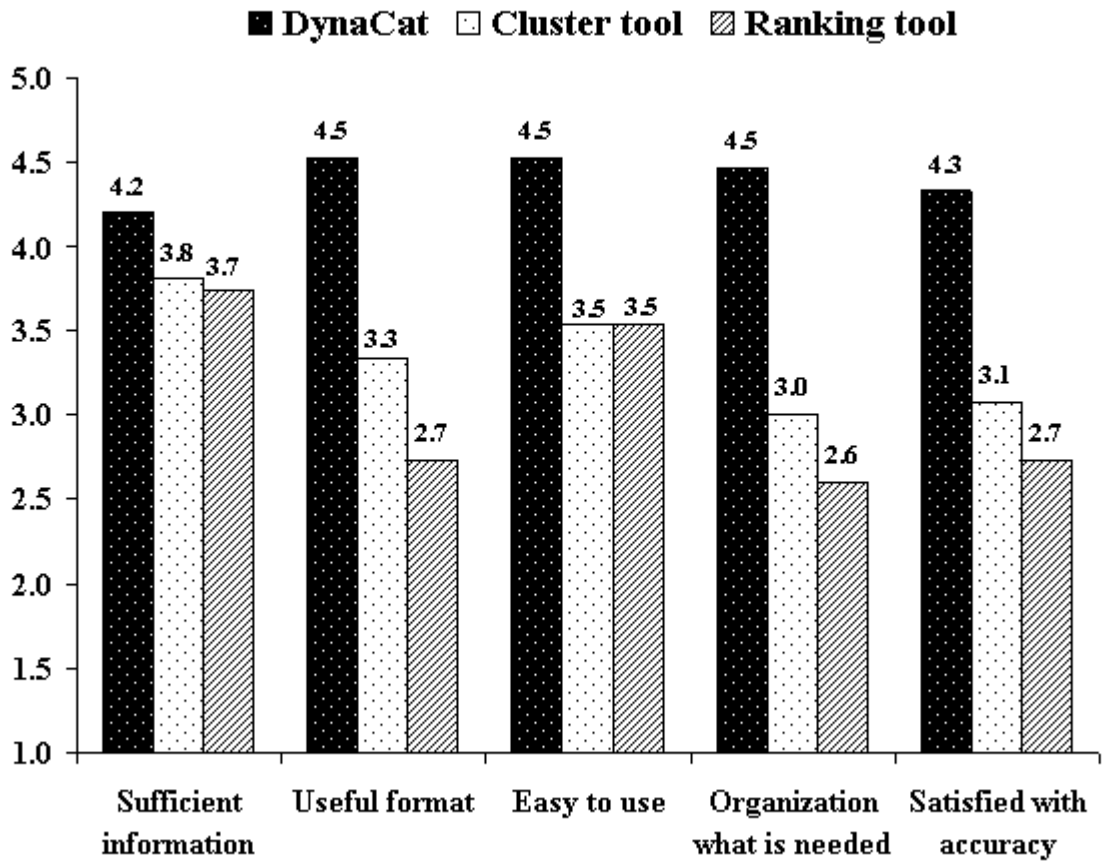


Figure 4.4 — Results from the second five questions from the validated user-satisfaction questionnaire. The mean values across all 15 subjects are shown on the *y* axis. The labels on the *x* axis show a brief summary of the questions asked — numbered 6 through 10. The full questionnaire is given in Appendix A. Subjects answered the questions using a scale from 1 to 5, where 1 meant *almost never* and 5 meant *almost always* (the ideal answer). The difference between DynaCat and the cluster tool was statistically significant ($p < 0.05$) for all five questions, as was that between DynaCat and the ranking tool, with the exception of question 6, about sufficient information — the *p* value for that question was 0.11.

from 1-to-5, where 5 was the most positive answer. The subjects' answers for DynaCat were significantly higher ($p < 0.05$) than those for either the ranking tool or the cluster tool, indicating that the subjects were more satisfied with DynaCat than they were with either the ranking tool or the cluster tool.

I created the remaining 16 questions on the questionnaire. For the first four questions, I provided statements and asked the subjects to rate them on a 1-to-5 scale where 1 meant

strongly disagree and 5 meant *strongly agree*. 5 was the ideal answer for three of those questions. The results are shown in Figure 4.5. For these questions, DynaCat also scored

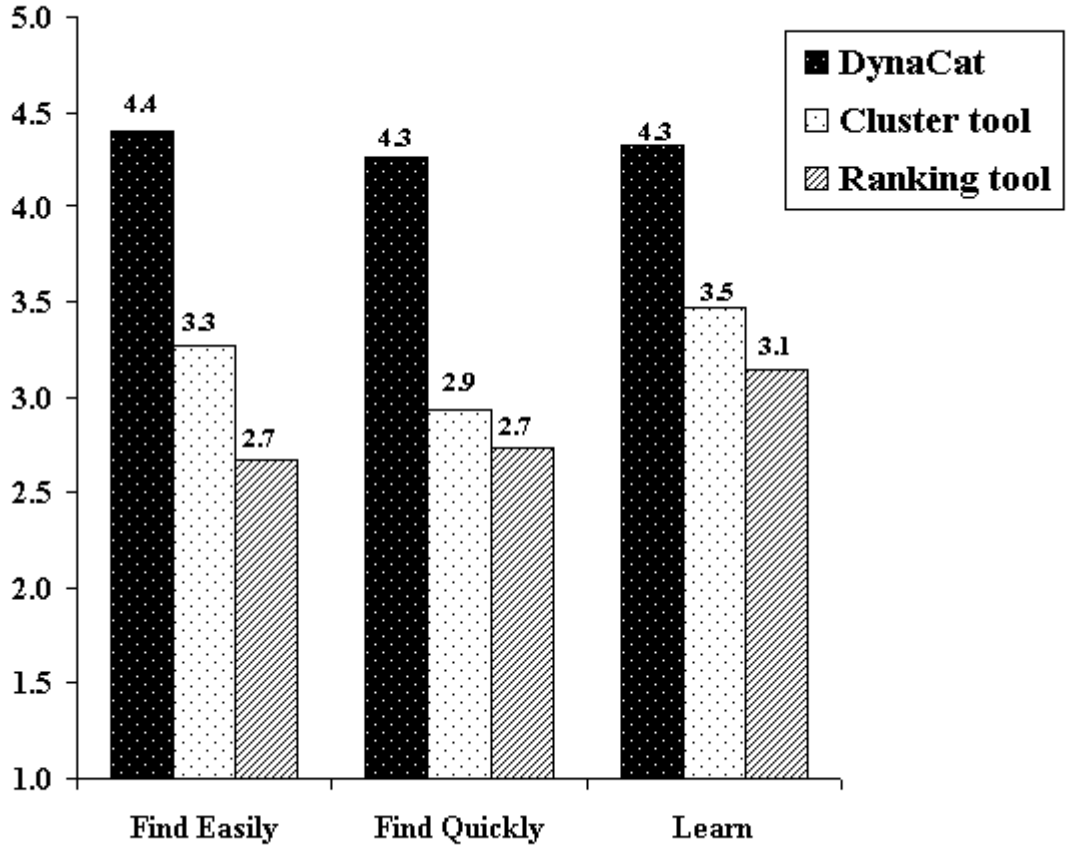


Figure 4.5 — Results for my user-satisfaction questionnaire. The mean values across all 15 subjects are shown on the *y* axis. The *x* axis shows a brief summary of the questions asked — numbered 11, 12 and 14. The full questionnaire is given in Appendix A. Subjects rated the statements on a scale from 1 to 5, where 1 meant *strongly disagree* and 5 meant *strongly agree* (the ideal answer). The difference between DynaCat and the cluster tool was statistically significant ($p < 0.01$) as were those between DynaCat and the ranking tool.

significantly higher than either the ranking tool or the cluster tool, indicating that the subjects found DynaCat better at helping them to find information quickly, to find information easily, and to learn about the topic corresponding to their query. Question 13 — *The amount of information provided in the search results was overwhelming* had an ideal answer of *1* (*strongly disagree*). For this question, the mean value that subjects assigned to DynaCat (2.40) was lower than those for the cluster tool (2.53) and for the ranking tool

(2.67), but the difference was not significant. The wording of this question, unlike that of all the other questions, does not refer to the system or to the organization of results; it refers to only the search results themselves. Thus, the subjects might have been answering the question based on how overwhelming the contents of documents were rather than how overwhelming the organization of those documents were.

The other 12 questions were either yes—no questions or open-ended questions. The results for the yes—no questions are shown in Figure 4.6. For these questions, DynaCat

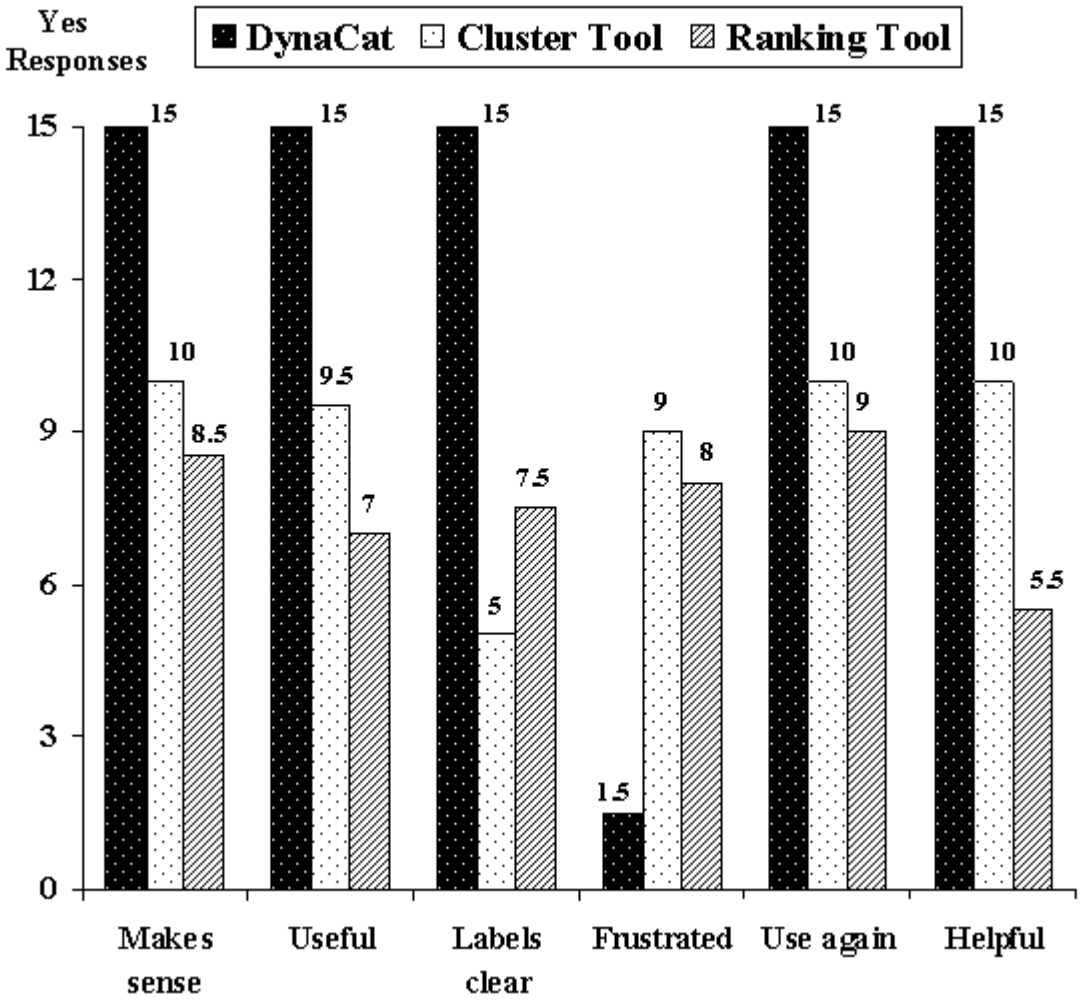


Figure 4.6 — Results of yes—no user-satisfaction questions. The *y* axis shows the total number of yes responses from each of the 15 subjects. The labels along the *x* axis show a brief description of questions 15, 17, 19, 22, 24, and 26. The full questionnaire is given in Appendix A. Some users answered *somewhat* instead of *yes* or *no*. Such answers were counted as one-half of a yes response.

also scored significantly higher than either the ranking tool or the cluster tool. Every subject agreed that the organization of documents by the category tool made sense, was useful, provided clear labels, and helped them to perform their tasks. For the cluster tool and the ranking tool, only two-thirds or fewer of the subjects answered those questions positively. Only one subject said that she found the category tool frustrating to use, and one other subject found it somewhat frustrating. Nine subjects found the cluster tool frustrating, and eight found the ranking tool frustrating. All 15 subjects said that they would use the category tool again when they wanted to search the medical literature; whereas only 10 subjects would use the cluster tool again, and only 9 would use the ranking tool again.

After the subjects finished using all the tools, I asked three more user-satisfaction questions:

- Which tool did you like best? (Ranking Tool, Cluster Tool, or Category Tool) Why?
- Which tool did you like least? (Ranking Tool, Cluster Tool, or Category Tool) Why?
- Did any of the tools help you learn more about the topic of the question? If so, which one?

The results for the final three questions appear in Figures 4.7, 4.8, and 4.9. Most subjects (87 percent) thought DynaCat helped them to learn about the answers to the question; whereas only 60 percent thought the ranking tool helped, and only 46 percent thought the clustering tool helped. Most people (70 percent) chose DynaCat as the best tool, and no one chose DynaCat as the tool that she liked the least. Subjects either really liked (23 percent) or really disliked (67 percent) the ranking tool and were more indifferent to the cluster tool.

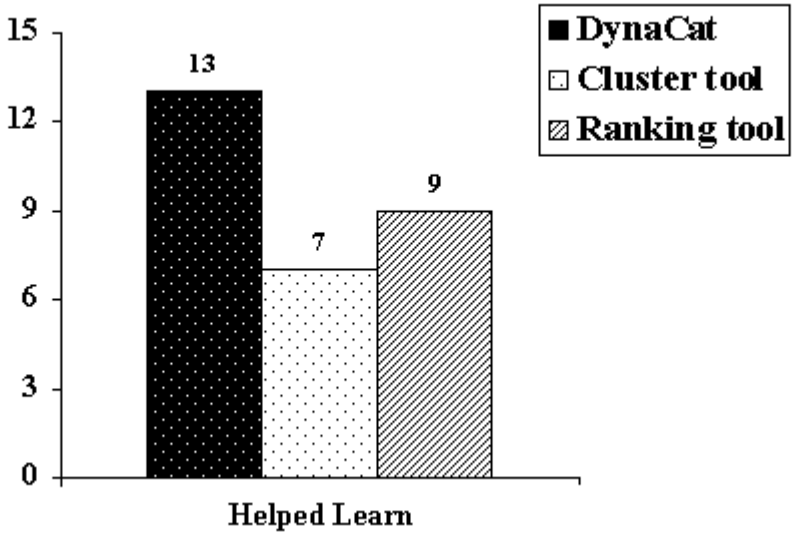


Figure 4.7 — Results for questions regarding which tools helped the subjects to learn. This chart shows the results of the final question: *Did any of the tools help you learn more about the topic of the question? If so, which one?*

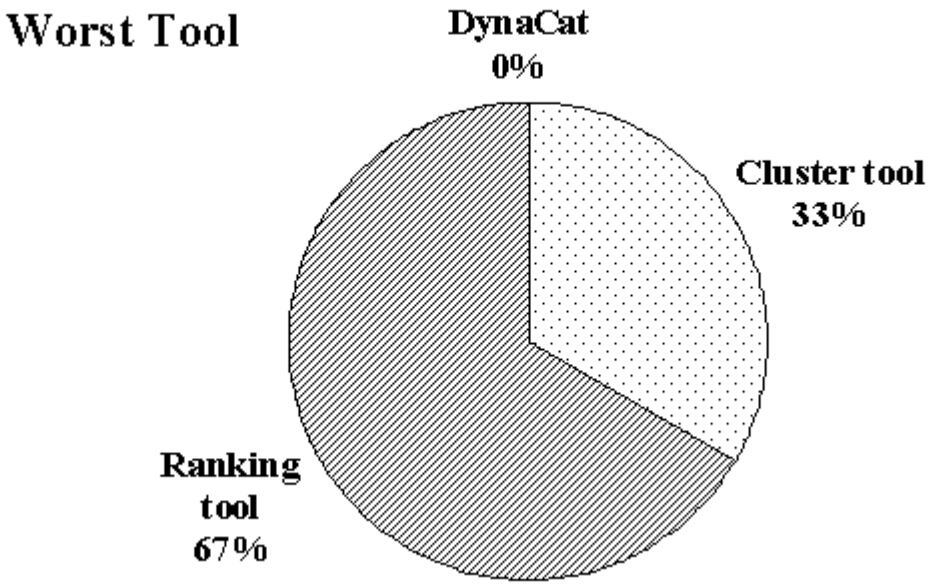


Figure 4.8 — Results for the question regarding the tool that subjects liked least. This chart shows the results for the question: *Which tool did you like least?* Most people chose the ranking tool as their least favorite. No one chose DynaCat as the worst tool.

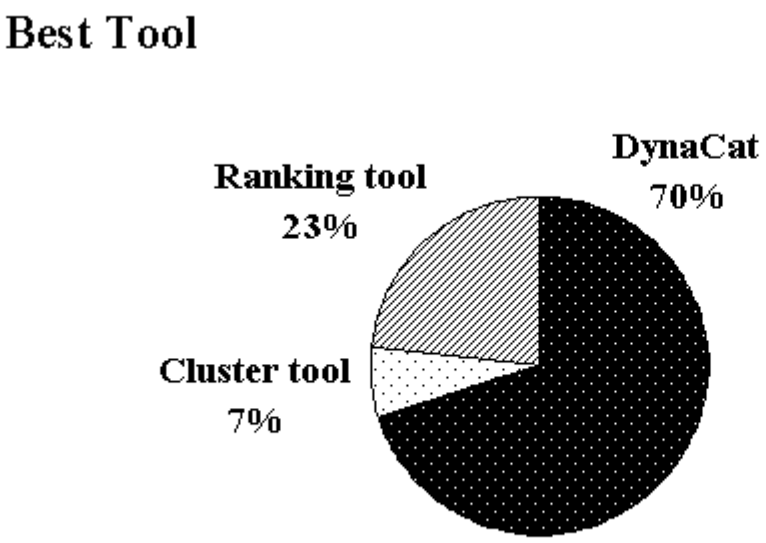


Figure 4.9 — Results for the question regarding the tool that subjects liked best. This chart shows the results for the question: *Which tool did you like best?* One person could not choose between the ranking tool and DynaCat, so I counted her answer as one-half of a vote for DynaCat and one-half of a vote for the ranking tool.

4.4.2.4 Comments and Answers to Open-Ended Questions

I asked several open-ended questions as part of the user-satisfaction questionnaire. It would be difficult to create a quantitative report of these results, but I have included several positive and negative quotes from the subjects in Tables 4.2, 4.3 and 4.4.

Table 4.2. Subjects' comments on DynaCat.

Positive Comments	Negative Comments
Clear and logical category names	Terminology was too technical
Hierarchy of categories	Want further classification of large categories
Alphabetic organization of categories	Did not like "Other" category
Easy to read and find specific information	
Articles grouped into manageable numbers	

Table 4.3. Subjects' comments on the cluster tool.

Positive Comments	Negative Comments
Better than no organization Easy to skim	Labels are not clear Labels don't match articles in cluster Not apparent how to find specific information Not intuitive

Table 4.4. Subjects' comments on the ranking tool.

Positive Comments	Negative Comments
Easy to understand the organization and browse Easy to look at more important info first Logical	Don't know how the ranking was done Seemingly random order No help in looking for specific information Waste time reading every title to find topics It can't know what I think are the most important documents

When the evaluation was over, three of the subjects asked whether they could look at more information using one of the tools. All three subjects asked to use the category tool.

4.5 Summary

In this chapter, I presented the pilot study and the final study that I conducted to evaluate DynaCat's performance. In these studies, I demonstrated that DynaCat is a more useful organization tool than is either a cluster tool or a ranking tool. The results showed that DynaCat is significantly better than the other two tools in terms of both efficiency in finding answers to the general question and of user satisfaction. The objective results for the amount learned were inconclusive; however, most subjects thought that DynaCat helped them learn about the topic of the query.

Chapter 5

Evaluation of Technical Claim

In Chapter 4, I demonstrated that DynaCat is more useful than either a clustering tool or a ranking tool. In this chapter, I discuss the evaluation of how well DynaCat creates categories and assigns documents to those categories. I present the objectives of this evaluation (Section 5.1), describe an early pilot study (Section 5.2), and report the final study (Section 5.3).

5.1 Objectives

In this part of the evaluation, my goal was to determine how well the system organizes the search results into a hierarchy of categories given the user's query type. This assessment includes determining how well the system

- Assigns meaningful labels to the categories
- Places documents in all appropriate groups
- Creates document groups that are responsive to the content and distribution of the documents in the search results

- Creates categories that correspond to the user's query

These desirable characteristics (Section 1.4) form the basis of my technical claim.

5.2 Pilot Study

The goal of this preliminary evaluation was to determine how well DynaCat places the documents in all and only the appropriate categories. I evaluated DynaCat using the query: *What are the complications of a mastectomy?* Figure 5.1 shows the web page that DynaCat generated using the keyword-pruning approach for that query. A search for *Mastectomy Adverse Effects* using the Oncology Knowledge Authority resulted in 92 different documents from CancerLit. The number of categories generated in the initial categorization was 53. If the system categorized the documents using every keyword of every document in the search result, the result would have been 263 different categories. In generating the hierarchical organization of categories (as described in Section 3.4.2), the system created 35 more categories for a total of 88 hierarchically organized categories. The maximum breadth of the hierarchy was 15; the maximum depth was 5.

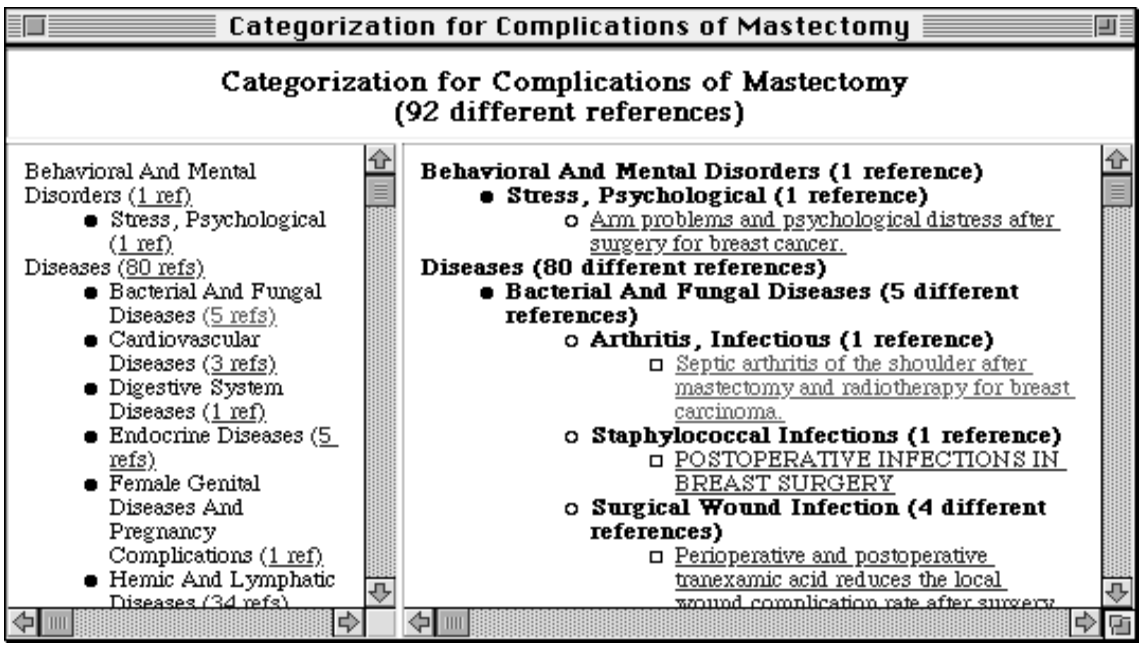


Figure 5.1 — DynaCat's interface for a search on the complications of a mastectomy.

To measure the accuracy of the categorization, I compared the categorization that DynaCat generated to one that a physician created. I randomly selected 30 documents from the original 92 search results and asked a physician to assign each document to one or more categories in the hierarchy of 88 categories generated by the system.

For each category, I determined the precision and recall of DynaCat using the physician's categorization as the gold standard. I defined precision as the number of documents that both the physician and the system assigned to the category divided by the number of documents the system assigned to the category. I defined recall as the number of documents that both the physician and the system assigned to the category divided by the number of documents that the physician assigned to the category.

The precision of my system, averaged across all categories, was 0.702, and the average recall was 0.440. Since there are no other systems that perform this exact task, it is difficult to use these figures in any comparison. However, a related task is the automatic classification of MEDLINE documents using keywords from the MeSH vocabulary. Yang and Chute evaluated several automatic classification approaches and found that the most precise approach had an average precision of 0.349 (Yang and Chute 1994a). They did not specify recall for their experiment results. Since the number of categories for my dynamic categorization task is much smaller than that for the task of assigning keywords to documents, it is not surprising that DynaCat provides a categorization with higher average precision.

This pilot study also assumed that the physician's categorization provided a good gold standard to measure against other means of categorizing documents. Most studies of the accuracy of clustering systems or classification systems make the same assumption, and use only one rater. However, document categorization is an inherently subjective task; people may not agree on which documents belong in which categories. For my pilot study, another physician may have categorized the documents differently, and the accuracy of one physician compared to that of another may not have been different from that of the system compared to a physician. I addressed this problem in the final study by comparing

the categorizations of several subjects to determine people's degree of agree in categorizing search results for a given query.

5.3 Final Study

In my final study, I tested my entire technical claim that DynaCat meets all four desirable characteristics (Section 1.4). I evaluated whether DynaCat placed documents in all appropriate groups by measuring how consistently subjects assign documents to categories, when they use an initial categorization structure that DynaCat generated, and how consistently DynaCat assigns documents to categories when compared to the subjects. I evaluated whether DynaCat satisfies the other desirable characteristics by asking the subjects to rate the characteristics of the categorization structure using a 1-to-5 scale, and to rate the same characteristics for the cluster structure that was generated using the cluster tool (Section 4.2.2). In the following sections, I explain the evaluation methods (Section 5.3.1), and report the study results (Section 5.3.2).

5.3.1 Methods

I outline the methods that I used for this evaluation in the following sections. I describe the data sets (Section 5.3.1.1), the subjects (Section 5.3.1.2), the procedure that these subjects followed (Section 5.3.1.3), and the evaluation metrics (Section 5.3.1.4).

5.3.1.1 Data Sets

To generate the documents for categorization, I used three specific queries: *What are the preventive measures for breast cancer?*, *What are the prognostic indicators for breast cancer?*, and *What are the diagnostic tests for breast cancer?* These queries represented three query types: *problem—preventive-actions.*, *problem—prognostic-indicators*, and *problem—tests*. I sent the queries to the Oncology Knowledge Authority (Tuttle, Sherertz, et al. 1994), which searched the CancerLit database. I limited the search to documents written in English that contained an abstract. For the first query, about prevention, the

search engine returned 83 documents that DynaCat assigned to 71 different categories. For the second query, about prognosis, the search engine returned 81 documents that DynaCat assigned to 69 different categories. The first two queries were used in the usefulness evaluation (Chapter 4), but to reduce the time it would take subjects to categorize the documents and thus make it easier to recruit subjects, I chose *What are the diagnostic tests for breast cancer?*, which I did not use in the usefulness evaluation, as the third query because the search engine returned only 44 documents. DynaCat assigned those documents to 27 categories.

5.3.1.2 Subjects

I chose to use physicians, rather than patients, as the subjects for this study because I assumed that physicians would be able to assess the content of the documents more thoroughly than most patients would. For the first set of search results, three physicians (two internists and one oncologist) categorized the documents. Two internists categorized the second set of search results, and four oncologists categorized the third set of search results. Each subject was paid to participate. Each completed a consent form.

5.3.1.3 Procedure

For this portion of the evaluation, subjects performed four tasks:

1. Read through the categories that were generated by DynaCat and the cluster labels that were generated by the cluster tool (Section 4.2.2).
2. Read the query, and the entire citation for each document in the search results.
3. Assign each document to all appropriate categories and to one of the clusters.
4. Rate the categories, and the clusters.

I gave all subjects a set of instructions (Appendix G), and asked them to read through both the hierarchy of categories (see example in Appendix H), and the clusters (see example in Appendix I) before they started to categorize the documents. So that they would know the context of the search results, I gave them the query.

I gave the subjects the hierarchy of categories that DynaCat generated for the query, but the categories did not contain references to the system-assigned documents. Giving subjects the system's categorization structure as a starting point may have biased them toward the system's categorization. However, if I gave no categorization structure to the subjects, they might not be have been able to generate a good categorization on their own. Creating a categorization structure requires abstract, analytical, and organization skills that different people have developed to different extents. It also requires more time and thought than does merely assigning documents to categories, and the subjects may not be motivated to spend the extra time to construct such a categorization structure carefully. Another problem with letting people create their own categorization structures is that there could be many ways to create a good categorization. Without an initial starting structure, the large differences in the chosen categories and the organization of those categories would inhibit comparing categorizations across subjects.

I instructed the subjects to examine each document and to determine the topics that were both discussed in the document and related to the given query. When they thought that a document was not relevant to the query, the subjects assigned the document to the category called *Not Relevant to Question*. Otherwise, the subjects put the document in every category that they thought represented the topics of the document. I explicitly instructed the subjects that they could assign a document to more than one category. If the subjects thought that a document belonged to a category that was not present in the provided structure, they could create their own category, label it, and assign the appropriate documents to that category.

5.3.1.4 Metrics

For this evaluation, I used both objective and subjective measures of categorization performance. I measured the categorization consistency across subjects, the consistency between the system's categorization and the subjects' categorization, and the accuracy of the system. I describe these metrics in Sections 5.3.1.4.1 through 5.3.1.4.3. Finally, I measured the subjects' assessment of the categorization through a short questionnaire (see Section 5.3.1.4.4).

5.3.1.4.1 Consistency Across Subjects

A statistic that calculates the proportion of agreement across subjects beyond the agreement due to chance is the kappa statistic (Cohen 1960):

$$\text{kappa} = \frac{P_{agree} - P_{chance}}{1 - P_{chance}}$$

where P_{chance} is the proportion of cases in which agreement is expected due to chance,
P_{agree} is the proportion of cases in which the subjects agree.

The original formulation of the kappa statistic, and most uses of the kappa statistic, were limited to cases where two raters assign one diagnosis each to each patient, which would correspond to two raters assigning one category each to each document. DynaCat assigns multiple categories to each document, and thus I cannot use the original formulation of the kappa statistic. Mezzich and his colleagues extended the kappa statistic to deal with situations where multiple raters assign multiple diagnoses to each patient, or, in this case, where multiple categorizers assign multiple categories to each document (Mezzich, Kraemer, et al. 1981). I used their formulation, where

$$P_{agree} = \frac{A}{A + J + K}$$

A is the number of categories assigned to the document by both raters (J and K).
J is the number of categories assigned to the document by rater J only.
K is the number of categories assigned to the document by rater K only.

The overall proportion of agreement (P_{agree}) for more than two raters is the average of the proportion of agreement for each pair of raters.

The proportion of chance agreement (P_{chance}) is the average of the proportion of agreement obtained between all combinations of raters and across all documents. The pseudo code for this calculation is show in Figure 5.2.

```
Initialize COUNT and SUM to 0
For each pair of distinct raters: R1 and R2
  For each pair of documents: D1 and D2
     Let SUM be SUM + (number of categories that were
     assigned both by R1 to D1 and by R2 to D2) divided by
     (number of categories assigned either by R1 to D1 or
     by R2 to D2)
     Increment COUNT
PCHANCE equals SUM divided by COUNT
```

Figure 5.2 — Pseudo code for calculating P_{chance}.

This calculation of chance agreement is based on the provided category assignments; it does not account for the chance agreement based on the number of categories from which the subjects can choose.

To interpret the kappa statistic, and to determine the consistency across the subjects, I used the benchmarks defined by Landis (Landis and Koch 1977), as shown in Table 5.1.

Table 5.1. Interpretation of the kappa statistic.

Kappa Statistic	Interpretation
< 0.0	Poor
0.00 – 0.20	Slight
0.21 – 0.40	Fair
0.41 – 0.60	Moderate
0.61 – 0.80	Substantial
0.81 – 1.00	Almost Perfect

5.3.1.4.2 Consistency Between System and Subjects

I calculated the agreement between the system's categorization and the subjects' categorizations using the same kappa measure as the one that I used for measuring the consistency among subjects. Ideally, the system should be as consistent with the subjects as they are

with one another. In other words, the kappa value from this calculation should be about the same as the kappa value from the across-subjects calculation.

5.3.1.4.3 Accuracy of the System

To measure accuracy, I compared the system's categorization to each subject's categorization. I created a contingency table of the categorization decisions for each category; see Table 5.2.

Table 5.2. Contingency table for the assignment of documents to a category[a].

	Documents that subject assigned to category	Documents that subject did not assign to category	Total number of documents
Documents that system assigned to category	*TP*	*FP*	*TP + FP*
Documents that system did not assign to category	*FN*	*TN*	*FN + TN*
Total number of documents	*TP + FN*	*FP + TN*	*ND*

a. *TP* is the number of true positives, which is the number of documents that both the subject and the system assigned to the category. *FP* is the number of false positives, which is the number of documents that the system assigned to the category that the subject did not. *TN* is the number of true negatives, which is the number of documents that both the system and the subject did not assign to the category. *FN* is the number of false negatives, which is the number of documents that the system did not assign to the category but the subject did. *ND* is the total number of documents in the search results.

Based on the contingency table, I could calculate any of the following metrics for each category:

$$\text{fallout} = \frac{FP}{TN + FP} = \text{false positive rate} = 1 - \text{specificity}.$$

$$\text{recall} = \frac{TP}{TP + FN} = \text{true positive rate} = \text{sensitivity}$$

$$\text{precision} = \frac{TP}{TP + FP}.$$

Because precision and recall are the standard metrics in the information-retrieval literature, I calculated pairwise precision and recall both among the subjects, and between the system and each subject. I averaged these metrics across all the categories for each set of search results.

5.3.1.4.4 Subjective Assessment of Categories and Clusters

To determine the subjects' assessment of the categories and clusters, I asked each subject to rate the following statements about the categories and the clusters:

- The labels on the categories (or clusters) are meaningful.
- The categories (or clusters) correspond to groups of documents that are appropriate for the citations provided.
- The categories (or clusters) correspond to groups of documents that are appropriate for the original query.

They were asked to use a 1-to-5 scale, where 1 corresponded to *almost never*, and 5 corresponded to *almost always*. The exact statements appear in the instructions in Appendix G.

5.3.2 Results

In this section, I discuss the results from the evaluation of my technical claim. I present the results of the consistency across subjects and between the subjects and DynaCat (Section

5.3.2.1), the accuracy across subjects and between the subjects and DynaCat (Section), and the subjective assessment of the categories and clusters (Section 5.3.2.3).

5.3.2.1 Consistency

To determine consistency among the subjects and between the subjects and the system, I calculated the average kappa statistic across all categories or clusters. The results for the categories are shown in Figures 5.3 through 5.5. The results for the clusters are shown in Table 5.3s.

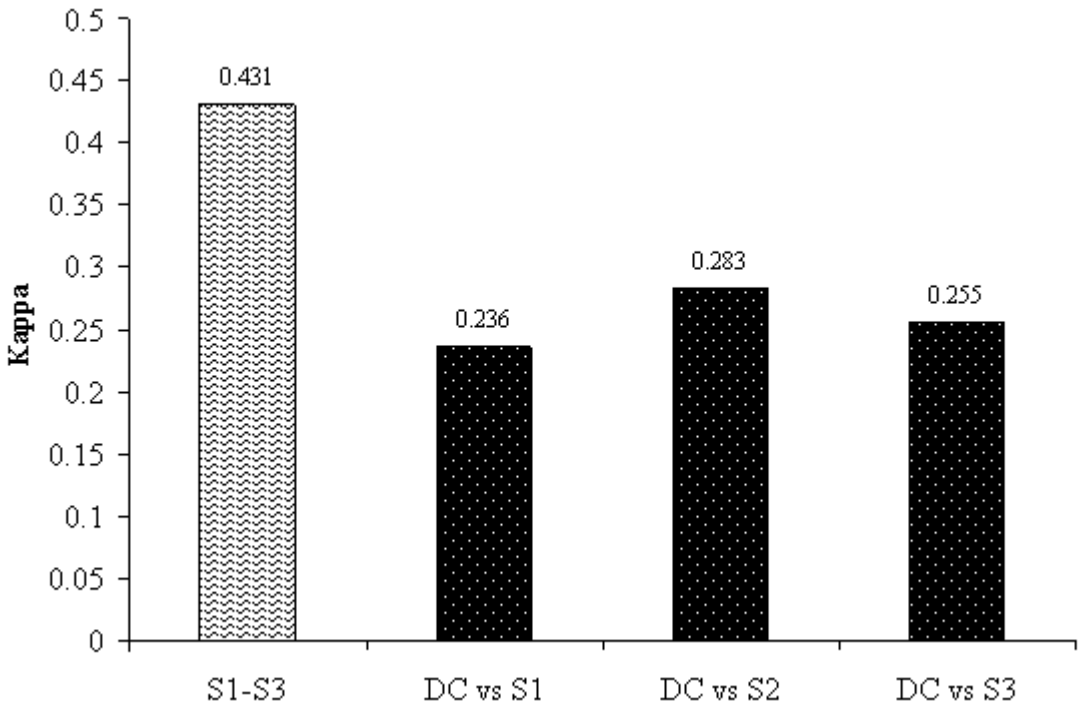


Figure 5.3 — Intercategorizer and DynaCat-categorizer consistency for the prevention-of-breast-cancer search results. The consistency in assigning documents to categories (any of 71 possible categories) across the three subjects (S1, S2, and S3) was moderate. The agreement between DynaCat and each of the subjects was fair.

For the categories, the consistency across the subjects ranged from fair to moderate, as did the consistency between DynaCat and the subjects. For the first query on prevention, DynaCat was somewhat less consistent with subjects than they were with each other, but this difference was small. Overall, DynaCat assigned documents to categories about as consistently as the subjects did. Although the consistency scores may not seem high, other

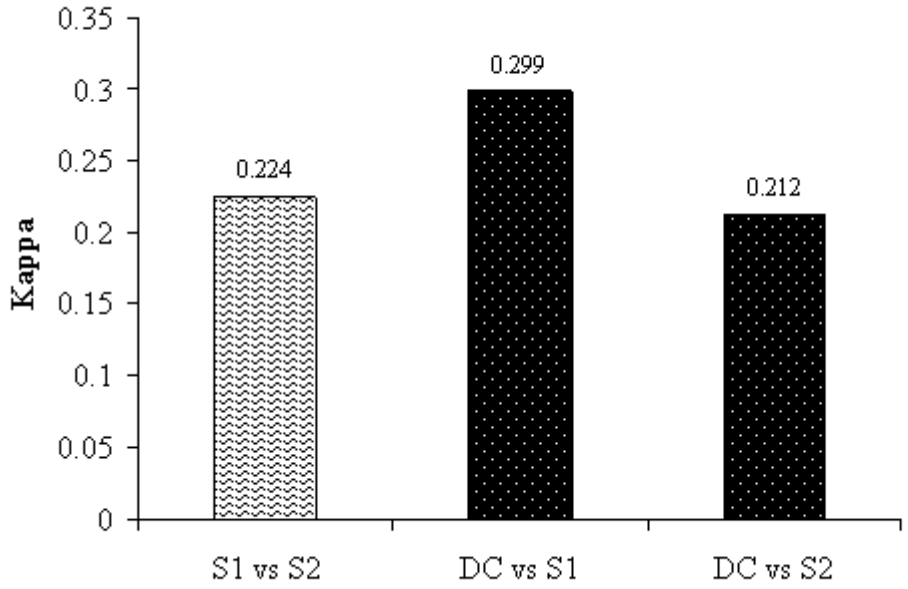


Figure 5.4 — Intercategorizer and DynaCat-categorizer consistency corresponding to the query on the prognostic indicators for breast cancer. The consistency in assigning documents to categories (any of 69 categories) between the two subjects (S1 and S2), and the consistency between DynaCat and each of the two subjects was fair.

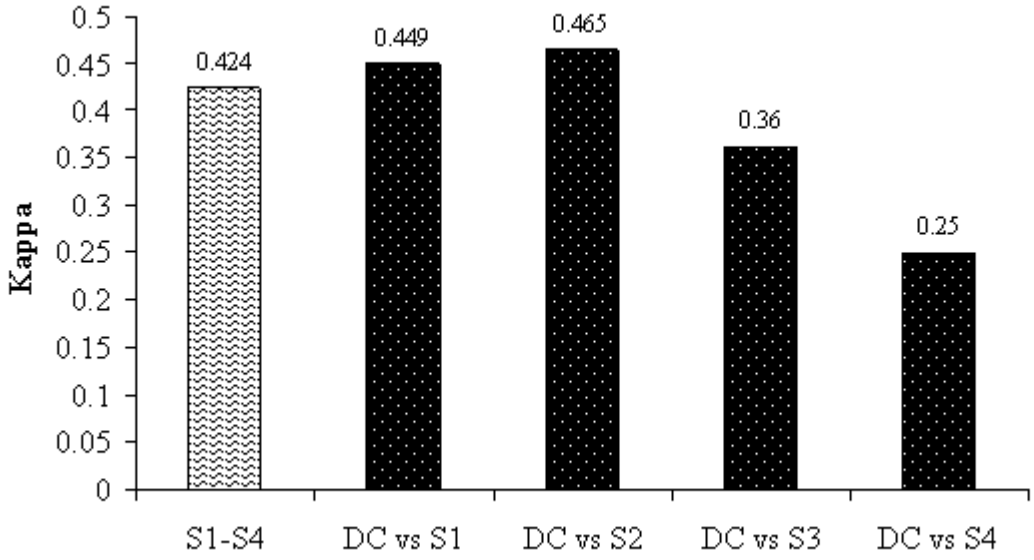


Figure 5.5 — Intercategorizer and DynaCat-categorizer consistency corresponding to the query on diagnostic tests for breast cancer. The consistency in assigning documents to categories (any of 27 categories) across the four subjects (S1, S2, S3, and S4) was moderate. The agreement between DynaCat and each of the subjects was fair to moderate.

studies have found that people often disagree on the assignment of index terms to documents (Ellis, Furner-Hines, et al. 1994). Even when the subjects are professional indexers, the consistency can be low. One study of MEDLINE indexers showed 33.8 percent agreement for all MeSH heading and subheading combinations (Funk and Reid 1983).

For the clusters, I also calculated the consistency across subjects and the consistency between the cluster tool and the subjects. The kappa scores were higher for the clusters

Table 5.3. Summary of consistency results for the cluster tool.

Query type	Average kappa across subjects	Average kappa between cluster tool and subjects
Prevention	0.529	0.637
Prognostic indicators	0.383	0.463
Diagnostic tests	0.735	0.619

(fair to substantial), than they were for the categories (fair to moderate). However, the probability of chance agreement for the clusters is substantially higher than that for the categories. For example, when there are five clusters (the largest number in this study), the probability of chance agreement is one-fifth (one divided by the number of clusters), because subjects were limited to choosing exactly one cluster. In contrast, the number of categories ranged from 27 to 71, and subjects were not limited to choosing only one category. Both of these factors dramatically decrease the probability of chance agreement for the categories.

I also compared the number of documents that the subjects assigned to the categories compared to the number of documents that DynaCat assigned to the categories. Table 5.4

shows the maximum and average number of documents assigned to a category by the subjects and by DynaCat. The numbers are comparable.

Table 5.4. Comparison of the maximum and average number of documents assigned to a category by the study subjects versus by DynaCat.

	Subjects		DynaCat	
Query type	**Max**	**Average**	**Max**	**Average**
Prevention	29	2.58	26	2.51
Prognostic indicators	45	3.30	24	2.67
Diagnostic tests	16	3.26	14	2.30

5.3.2.2 Accuracy

To determine each system's categorization accuracy, I calculated precision and recall for each of the subject's categorizations compared against each of the other subject's categorizations, and for the system's categorization compared against each of the subject's categorizations. The average results for the categories are shown in Figure 5.6; the average results for the clusters are shown in Figure 5.7. I included only one value for the subjects' precision and recall because average precision is equal to average recall when every subject acts as the gold standard for one round of precision and recall calculations. This fact becomes more obvious if you examine the contingency table (Table 5.2). The number of false negatives when subject A's categorization is compared against subject B's categorization is the same as the number of false positives when subject B's categorization is compared against subject A's categorization.

For the prevention query, and the diagnostic tests query, DynaCat's average precision and recall were slightly lower than the subjects, but its scores were well within one standard deviation of the subjects. Overall DynaCat's accuracy is comparable to that of the subjects. The cluster tool's accuracy also is comparable to that of the subjects.

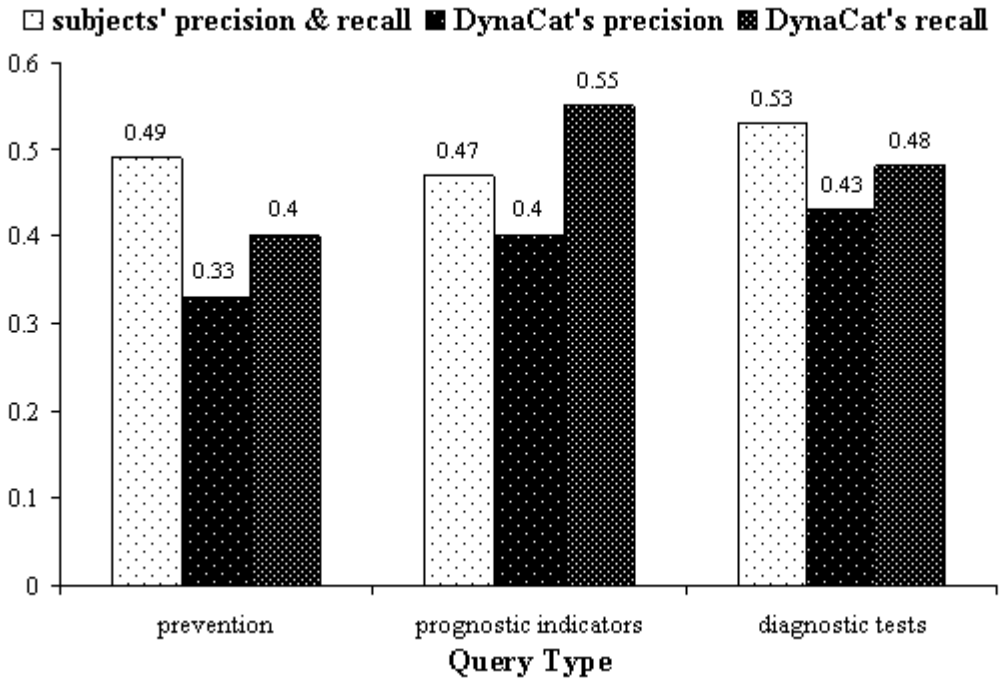


Figure 5.6 — Average precision and recall in comparisons of DynaCat to the test subjects, and the subjects to each other.

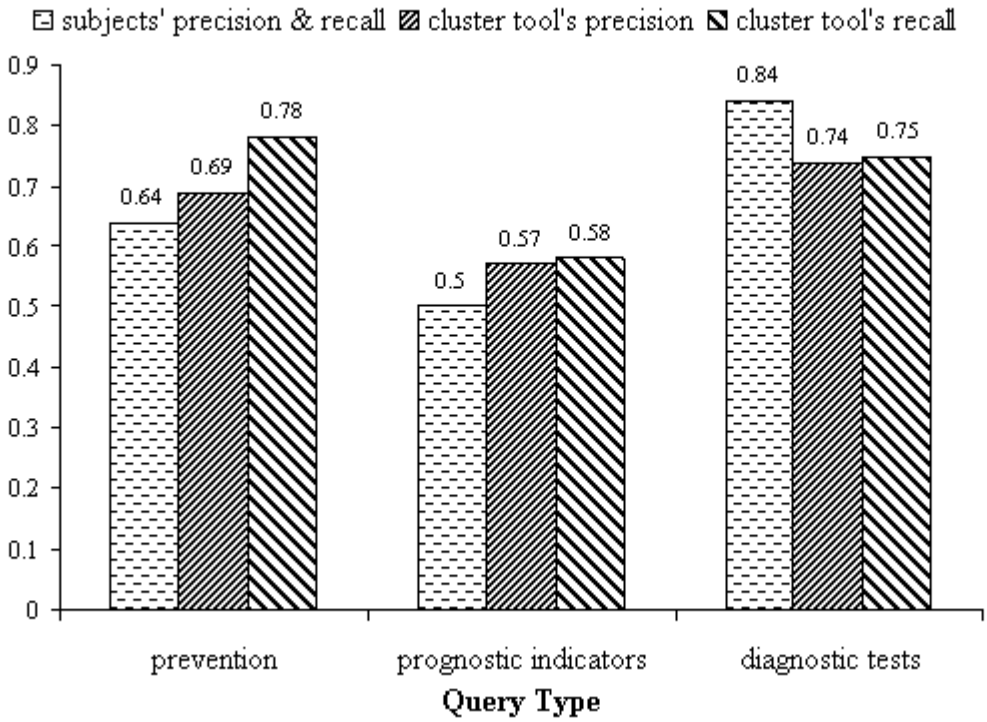


Figure 5.7 — Average precision and recall in comparisons of the cluster tool to the test subjects, and the subjects to each other.

5.3.2.3 Subjective Assessment

Six subjects completed the assessment of the desirable characteristics for the categories and clusters. The average of their scores for each characteristic is shown in Table 5.5. The three subjects for the first query, about prevention, did not assess the characteristics of the categories and clusters because their answers may have been biased: They knew me, and may have been able to determine which system was mine. For all three characteristics, the mean score for the categories was greater than 3, and was greater than the mean score for the clusters. These results provide evidence that DynaCat assigns meaningful labels to the categories, creates categories that correspond to the search results, and creates groups that correspond to the query. However, the difference between the categories' scores and the clusters' scores was not statistically significant. To determine whether subjects think that DynaCat performs these tasks significantly better than does the cluster tool, we would have to run a study with more subjects.

Table 5.5. Average scores for the subjects' assessment of the desirable characteristics for categories and clusters[a].

Characteristic	Categories score	Clusters score
Meaningful labels	3.67	3.33
Groups correspond to search results	3.83	3.50
Groups correspond to query	3.33	2.83

a. Subjects used a scale from 1 to 5, where 1 meant *almost never* and 5 meant *almost always*.

5.4 Summary

In this chapter, I presented the pilot study and the final study that I conducted to evaluate DynaCat's technical performance. In these studies, I demonstrated that categorization by DynaCat was about as consistent with the physicians' categorizations as the physicians's categorizations were with each other. In the subjective assessment of the categories and

clusters, physicians rated the categories higher than the clusters in terms of how meaningful the labels were, how well the categories corresponded to the query, and how well the categories correspond to the documents in the search results, although none of these differences were statistically significant. In Chapter 6, I summarize the contributions of my thesis research, the limitations of my current approach, and the possibilities for building on this research in the future.

Chapter 6

Summary and Conclusions

In this chapter, I summarize dynamic categorization (Section 6.1), discuss the contributions of my research (Section 6.2), report on the limitations of my current approach (Section 6.3), and present avenues for future work (Section 6.4).

6.1 A Knowledge-Based Approach to Organizing Search Results

This dissertation offers a new, knowledge-based method for dynamically categorizing search results. I presented, DynaCat, a system that implements this approach for the domain of medicine. DynaCat uses knowledge of the user's query and a model of the domain terminology to generate query-sensitive summaries of the kinds of information found in the search results.

I explained how DynaCat provides information about (1) what kinds of information are represented in (or are absent from) the search results, by creating document categories with meaningful labels and by hierarchically organizing the document categories; (2) how

the documents relate to the query, by making the categorization dependent on the type of query; and (3) how the documents relate to one another, by grouping ones that cover the same topic into the same category.

The technical evaluation demonstrated that the categorization generated by DynaCat was about as consistent with the physicians' categorizations as the physicians's categorizations were with each other. These results suggest that DynaCat creates reasonable document categories and assigns documents to categories appropriately.

The usefulness evaluation showed that users could find more answers in a fixed amount of time, and were more satisfied with their search experience when they used DynaCat than when they used either the cluster tool or the ranking tool. Users indicated that DynaCat provided an organization of search results that was clear, easy to use, accurate, precise, and helpful. They thought that DynaCat helped them to find answers easily and quickly, and to learn about the information related to their query.

Because the studies involved a small number of queries in one domain, more evaluation is needed to justify broader claims. Nevertheless, these initial results suggest that, by using knowledge about users' queries, and the kinds of organizations that are useful for those queries, DynaCat can provide users with satisfactory search experiences.

6.2 Contributions

The primary contribution of my work is to the interdisciplinary field of medical informatics. My work expands on ideas from the contributing fields of information access, and knowledge-based systems, to create a useful tool for the domain of medicine. I elaborate on these contributions in Sections 6.2.1 through 6.2.3.

6.2.1 Information Access

The main contribution of my research is in creating a new approach to organizing search results that helps users who have general queries to gain a high-level understanding of their search results and to identify quickly answers to their queries. This approach organizes documents into a hierarchy of categories, and automatically

- Assigns meaningful labels to the categories
- Places documents in all appropriate categories
- Creates categories that correspond to the content of the documents in the search results
- Creates categories that correspond to the user's query

No previous approach to organizing documents (either relevance ranking or clustering) provides all these abilities. My approach provides these capabilities because it is based on a representation of the documents that is semantically richer than the typical vector-space representation.

My usefulness evaluation also provided insight into what users liked and disliked about the three different approaches to organizing search results. For example, when they assessed the cluster tools users rated the clarity of the cluster labels and the labels' correspondence to the search results as poor. As another example, when users assessed the ranking tool, they complained that they did not understand how the ranking was done. This information could be used by cluster, and ranking system developers to create tools that provide a more satisfying and useful search experience.

6.2.2 Knowledge-Based Systems

Much of the research in information access emphasizes statistical techniques, rather than knowledge-based approaches. Two reasons for the prevalence of statistical techniques are the amount of work required to construct and maintain the necessary models for knowledge-based approaches, and the lack of evidence that such approaches perform better than

the statistical approaches. In this dissertation, I described a knowledge-based approach that builds on existing domain models, thus reducing the creation and maintenance effort. I also demonstrated that this approach provides a more useful environment for exploring search results than either of the common statistical approaches, relevance ranking and clustering. One of my goals is that my results encourage other researchers to pursue knowledge-based approaches to information access.

6.2.3 Medicine

I undertook this research because I want to improve the ability of patients and medical professionals to access the vast quantity of medical information. Even when you consider only the primary medical literature, the amount of information can be overwhelming. MEDLINE alone contains more than 8.6 million bibliographic citations and author abstracts from over 3800 current biomedical journals and adds 31,000 new citations each month. My evaluation clearly indicated that DynaCat provides one way to help people understand their search results and thus to find answers to their questions quickly.

6.3 Limitations

In this section, I present the limitations of my research in the scope of the query model used (Section 6.3.1), and in the effort required to create the domain models (Section 6.3.2).

6.3.1 Scope of Query Model

DynaCat categorizes search results from only those queries that map to one of the query types in the query model. My current query model is not an exhaustive model of medical queries. My goal was to create a proof of concept that the dynamic categorization method can be applied to a variety of query types, rather than to demonstrate the comprehensiveness of the model. Future work could explore extensive modeling of medical queries.

Another option is to allow users to categorize their search results using any of the predefined category types when the query model does not cover their query. Without query information, the system may not be able to generate a categorization with the same quality as it could with the query information, but the categorization could still be more useful than the alternative organizations, such as relevance ranking.

6.3.2 Domain-Modeling Effort

Dynamic categorization is a knowledge-based approach to organizing search results; thus, it requires that the system developer create the appropriate models. Dynamic categorization requires two types of domain models: a terminology model and a query model. The construction, use, and maintenance of these domain models can be time consuming and difficult for system designers. To reduce the modeling burden, I used an existing domain model from the National Library of Medicine as the terminology model. In Section 3.2.2.1, I describe several other terminology models that could be adapted for use in dynamic categorization for other domains; however, even if a system designer uses existing terminology models, she must learn about that terminology model, and must make connections to the query model. She must also construct a query model for the domain of interest. However, I justify this extra demand by demonstrating that systems can use my knowledge-based approach to generate organizations of search results that are more useful than those we can obtain using domain-independent, statistical approaches, such as clustering and relevance ranking.

6.4 Future Work

My research on dynamic categorization provides the basis for a series of research projects on knowledge-based techniques for improving access to medical information. In Sections 6.4.1 through 6.4.3, I discuss several such projects.

6.4.1 Interactive Categorization Environment

In my current approach to categorization, the system infers which types of categories are of interest based on the user's query and categorizes the documents into only those types of categories. However, users may want to categorize the documents along other dimmensions. An alternative approach would be to provide an environment for helping the user choose which types of categories are appropriate for their needs, the matching documents, and the query. Such an interactive categorization environment would allow the users to select subsets of individual documents or categories of documents, and recategorize them in different ways, such as according to study quality or subject characteristics.

6.4.2 Information Filtering

Dynamic categorization could be adapted to information-filtering tasks, where a long-standing query is specified and matched against newly published information. The first step in the filtering process could proceed as usual where the system selects documents that match the user's standing query. As in dynamic categorization, the system then could identify the domain-specific terms and their semantic types that are present in the filtered documents. Using this enhanced representation of the documents, the system could recommend various categorization options based on the filtered documents and the user's standing query.

Such a tool could be particularly useful as a maintenance tool for web sites of frequently asked questions (FAQs). An example is the National Cancer Institute's CancerNet web site, which has FAQs that provide summaries of, and pointers to related articles in the medical literature (NCI 1998).

6.4.3 Categorization of Informal Medical Information

Currently, DynaCat categorizes only medical journal articles, but the general methodology also should be applicable to informal medical information such as that found on web pages. With the web's current unannotated state, the keyword-pruning categorizer

approach could not work, because few web pages contain keywords that represent the page's content. However, researchers have proposed several options such as XML, RDF, and MDEF that would make it easier for web developers to provide structured, semantic information about their web pages. If any of those approaches dominate the web, I could modify DynaCat to take advantage of this semantic information, similar to the way it uses keywords now.

A second option would be to use the information-extraction categorizer on web pages or other unannotated information sources. Currently, this method is not scalable because it requires many extraction templates for each possible question, and it is too time consuming to create the necessary extraction templates manually. However, I may be able to extend the current research in semi-automatic generation of extraction templates (Riloff 1993; Riloff 1996a; Riloff 1996b). The current approaches generate extraction templates based on many examples for a specific query, but DynaCat needs extraction templates for query types, rather than individual queries. If I can extend the current approaches to generate extraction templates for abstract query types, the information-extraction categorizer could become a reasonable option.

6.5 Concluding Remarks

The amount of medical literature continues to grow as the content becomes increasingly specialized. At the same time, many patients and their families are becoming proactive in searching the medical literature for information regarding their medical problems. Medical journal articles can be intimidating for lay people to read; thus they need tools to help them to sift through and to understand the information that they seek. I have described an approach to organizing medical search results and have proved that my approach is helpful; my research should lead to tools that will help lay people—both patients and their families—to explore the medical literature, to become informed about health-care topics, and to play an active role in the decisions about their own medical care.

Health-care workers also need tools to help them cope with the vast quantities of medical information that they must access to care for their patients, and to further medical research. Although my system was evaluated with only patients as users, it could be used by health-care workers as well. The questions that health-care workers ask may be more specific or more varied, but the terminology used and categorization process would remain the same. If tools based on my research were available to health-care workers, they might be able to find the needed information fast enough during a patient visit, when that information is most useful.

With the explosion of information available to consumers on the web, the general public faces similar overload problems for many types of information. As I argued in Section 3.3.1.1, my general approach to knowledge-based organization could be extended to other domains. Such research could result in new tools that would help all users to explore quickly and effectively the information space related to their individualized needs.

Appendix A

User-Satisfaction Questionnaire

Using the scale below, please answer questions 1-10:

Scale:

1 = Almost never
2 = Some of the time
3 = Almost half of the time
4 = Most of the time
5 = Almost always

1) Is the organization of the information clear?

2) Does the system provide the precise information you need?

3) Is the system accurate in assigning documents to categories?

4) Is the system user-friendly?

5) Does the organization of the information content meet your needs?

6) Does the system provide sufficient information?

7) Do you think the information is presented in a useful format?

8) Is the system easy to use?

9) Does the system provide an organization of the information that seems to be just about exactly what you need?

10) Are you satisfied with how well the system assigns documents to categories?

Using the scale below, please answer questions 11-14:

Scale:

1 = Strongly disagree
2 = Disagree
3 = Uncertain
4 = Agree
5 = Strongly agree

11) The organization of the search results makes it easy to find information.

12) The organization of the search results makes it easy to find information quickly.

13) The amount of information provided in the search results was overwhelming.

14) The organization of the search results made it easy to learn about information related to the query.

Please answer the remaining questions in your own words:

15) Does the organization of the documents make sense?

16) How do you think the organization could be improved?

17) Do you find the organization useful?

18) If so, in what way?

19) Do the labels that describe each group of documents make sense?

20) What do you like about the organization of the documents returned?

21) What do you not like about the organization of the documents returned?

22) Were you frustrated when you used the system?

23) If so, why?

24) Would you use the system again when you want to search for medical information?

25) Why or why not?

26) Did the grouping of the documents help you perform your tasks?

A p p e n d i x B

Frequently Asked Questions About Breast Cancer

Prevention

1. I have a mother (and/or sister) who has been diagnosed with breast cancer. Should I have a double mastectomy to prevent myself from also getting breast cancer?

2. Almost all the women in my family, on both sides, have had breast cancer. Should I have a double mastectomy to prevent myself from also getting breast cancer?

3. *I have no family history of breast cancer, so why should I worry?

4. How can I prevent breast cancer with diet and vitamins?

5. Now that I have breast cancer, I'm worried about my daughter. How can she prevent breast cancer in her own body?

6. If I have children while I am still young (under 35), can I prevent breast cancer?

7. "I did everything right. Why did I get breast cancer?" ("Everything right" means low-fat diet, exercise, children under 35, organic foods, no drinking or smoking, etc.).

8. Did (emotional) stress cause my cancer?

9. Is cancer contagious? Can I catch it from someone or can someone catch it from me?

Screening and Detection

10.*What are the screening guidelines?

11.I am in my 30s (40s/50s). Should I be getting mammograms? How often?

12.I have a lump. What type of biopsy would be the most accurate for me?

13. Some calcifications showed up on my mammogram. What type of biopsy would be the most accurate for me?

14.I have a painful lump in my breast. My friends tell me that it can't be cancer because cancer never hurts. Is this true? Should I have it checked by a doctor?

15.*I have found a lump in my breast. My physician also recognizes it as suspicious but recommends that I wait for six months and watch it. I'm very scared. What should I do?

16.*What is the doctor looking for when she/he says to "watch and wait"?

17.I am in my 20s (30s/40s) and have a lump. My doctor tells me that I am too young to have cancer and that it is probably just a cyst. He does not want to do anything about it, but I am still worried. What should I do? Could I have breast cancer?

18.I have been diagnosed with breast cancer at a somewhat (or very) advanced stage. I have been having regular check-ups for many years. Why didn't this ever show up in my physical exams or on my mammograms?

19.*I know that there are three parts to early detection, breast self exam, clinical exam and mammograms. If I can't find a lump and my doctor can't find a lump, am I certain to be safe? (Another version of this question is, if I find a lump, but it doesn't show up on the mammogram and my doctor doesn't feel it, am I safe?)

20.*Why does the mammographer take two pictures of each breast? (Or, my physician only takes one photo of each breast, but my friends get two pictures of each breast taken. Why?)

21.*Is it true that I should remove all deodorant and bath powder before going in for a mammogram? Why?

22.*What is the radiation dosage for a mammogram? How does this compare to other procedures or activities? Can't the mammogram itself cause cancer?

23.*If my mammograms are not easy to read, are there any other screening procedures I can try?

24.*Does prior breast surgery or implants affect the reading of the mammograms?

25.*Why does it ahve to hurt so much when I have a mammogram? Is there a way to lessen the pain?

Diagnosis & Prognosis

26. What are the different types of biopsies?

27.I have calcifications. What would be the most accurate type of biopsy for me?

28.I have a lump. What would be the most accurate type of biopsy for me?

29. Does having a biopsy raise a woman's chance of getting breast cancer?

30. If I have a biopsy and later want to breast feed, will I be able to ?

31. Where can I find the best doctors for my treatment?

32.*I have had a fine needle aspiration that indicates that I have cancer. My physician is recommending that I go in for surgery and an excisional biopsy all during the same procedure. He says that he can use the "frozen sections" to determine how far the cancer has spread and decide whether to do a mastectomy or a lumpectomy while I am on the operating table. This sounds like a good idea to me because I save myself from a second surgery. Is this a good plan?

33.*Should I get a second opinion? How should I choose that physician?

34.*What is a tumor board?

35.*How long does the biopsy take? Will I have to stay overnight in the hospital?

36.*What is a needle localization? Does it hurt?

37.*When will I find out what the results are? Who will tell me?

38.*Will I have a huge scar? Where will the scar be?

39.*When will I be able to return to work or continue my normal routines?

40.*If the results are positive, do I need to hae surgery and other treatments immediately? How long can I take to decide what to do?

41. How long after the surgery should I begin chemotherapy?

Surgery

42.*How do I choose a surgeon?

43.*Should I have a mastectomy or a lumpectomy?

44.*What are the side effects, risks and possible complications of each surgery?

45. Can I have a recurrence if I have a mastectomy?

46.(Mastectomy is often referred to as masectomy and recurrence is often called re-occurence.)

47.*How does a recurrence affect my survival?

48.*What is the differnece between long-term and short-term survival?

49. How big will the scar be? Where will it be? How long will it take to heal?

50. Will the scar tissue from a lumpectomy or a biopsy interfere with follow-up mammograms?

51. Do I have to have radiation if I have a lumpectomy?

52. Do I have to have radiation if I have a mastectomy?

53.*Since my doctor does not feel any enlarged lymph nodes under my arm, why is she/he recommending that I have some of my lymph nodes removed?

54. If the doctor removes my lymph nodes, will he have a better chance of getting all the cancer out?

55. How many lymph nodes do I have? How many have to be removed? Why not remove them all?

56. Does everyone need to have their lymph nodes removed? Do I?

57. Why can't I keep the nipple when my breast is removed?

58. Prior to surgery, patients often want precise descriptions of what will happen in surgery and also want to see pictures of expected results. They also often want to talk to other people who have had specific types of treatment to know what to expect both during and after surgery and how to deal with side effects and consequences of surgery.

59.*Should I have local or general anesthesia?

The Pathology Report

60.I have had a biopsy and now see all these strange words on the pathology report? What do they mean? (There is a long list of words that have no meaning to the newly diagnosed patient. Patients need to know not only the definitions of the words, but the implications of the words and phrases and what the different combinations of prognostic indicators mean.)

61.I thought there was only one type of breast cancer. Now I see that I have a specific type of breast cancer and that there are other types of breast cancer. How does my type of breast cancer fit into the overall picture?

62. What type of breast cancer do I have?

63. How long will I live?

64. What if I do nothing?

65.*Can the pathologist tell how fast the cancer is growing?

66.*How can the pathologist know whether or not all the cancer has been removed?

67.*What is the difference between infiltrating and invasive? Invasive and in-situ? (All these terms are often confusing. Patient needs to know that there is a difference between invasive and in-situ disease and that there are degrees of invasion).

68.*What are estrogen and progesterone receptors and what do the numbers on my pathology report mean to me in terms of treatment?

69.*Why does my menopausal status affect my treatment options?

70.*Am I less likely to have recurrence if I have my surgery done in the latter half of my menstrual cycle?

Reconstruction

71. Should I have immediate reconstruction or delayed reconstruction?

72. Can I have mammograms after having a tram flap?

73. Can I have mammograms after having an implant?

74. Can the cancer recur in the tram flap tissue?

75. Can the cancer recur beneath my implant?

76. Again, patients want precise descriptions of the procedures and techniques and want to see photographs.

77. What does a reconstructed breast feel like? (To me, not my husband or doctor).

78. What if I gain or lose weight after reconstruction? Will my reconstructed breast (implant/tram flap) no longer match the opposite breast?

79.*What is the difference between a saline and a silicon implant?

80.*Is there a danger of the implant causing auto-immune disease?

81.*What is the difference between a tissue flap and an implant?

82.*What are the risks of infection, leakage or rupture of an implant?

83.*Will I have perfect symmetry after reconstruction?

Radiation

84. Will radiation increase my risk of getting cancer elsewhere?

85.*Will radiation damage my heart, bones, lungs, reproductive system?

86.A question not often asked, but that should be asked:

87. If I have radiation now, and have a recurrence later, will they be able to irradiate for that tumor also?

88. Will I have radiation burns?

89.*Will the effects of the radiation treatment make future mammograms more difficult to read?

90. How do I care for my skin during my radiation treatments?

91. Why do I get weekends off from my radiation treatments?

92. Why can't we just do it all at once?

93. What is a boost?

94. Should I have my lymph nodes irradiated?

95. Can I exercise during treatment?

96.*Now that I have had my radiation treatment, my breast feels more dense and rubbery. Will this go away?

Chemotherapy

97. How long after the surgery should I begin chemotherapy?

98.*What should I do to prepare for chemotherapy? (go to the dentist, etc.)

99. Will I lose my hair? Will I lose it all at once? Will it grow back in gray?

100. Does chemotherapy cause heart damage?

101. Will chemotherapy put me into menopause?

102. Does chemotherapy cause other types of cancer?

103. Will I be sick? (or, during treatment why do I feel so sick and fatigued?)

104. Will I be hospitalized during treatment?

105. Will I be able to work while undergoing chemotherapy?

106. How should I deal with nutrition while I am undergoing treatment?

107. How can I boost my immune system while undergoing treatment?

108. What are the signs of infection that I should watch for during chemotherapy?

109. Will a bone marrow transplant increase my odds of survival?

110. Is CMF or CAF more effective?

111. Can I have chemotherapy and radiation at the same time, or do I first have one and then the other?

112. Why do some people have chemotherapy and radiation before surgery?

113. Can I exercise during treatment?

114.*How long will each chemotherapy session take, how often will I be treated, and how long will I continue to have chemotherapy treatments?

115.*If I have to skip or postpone a session, will the therapy not be as effective?

116.*If I have a recurrence, will I have chemotherapy again? The same type?

117. Should I begin my tamoxifen before I start radiation, or should I wait until I have completed radiation?

118. Does tamoxifen cause menopause?

119. How long can I continue taking tamoxifen?

120.*Should I take my two tamoxifen pills at the same time, or should I take one in the morning and one in the evening?

121.*Tamoxifen has caused severe hot flashes for me. How can I reduce the discomfort?

122.*Will tamoxifen make me infertile?

123. What are the side effects of tamoxifen?

124.I'm taking tamoxifen and feel depressed. I think the tamoxifen is causing my depression. My doctor says I am naturally depressed because I have been diagnosed with cancer and that the tamoxifen has nothing to do with it. Is he right?

125. Can I take tamoxifen if I am estrogen receptor negative?

126. Is tamoxifen effective on pre-menopausal women?

127.I've been taking tamoxifen for several years and now am having vision problems. What doctor should I talk to about this? Could it be the tamoxifen, or am I just getting old?

128. Is there really such a thing as "chemo-brain"? (Fuzzy thinking, forgetfulness caused by chemotherapy). How long does it take to go away?

129. How can there be such a thing as "chemo-brain" if the chemotherapy does not cross the blood/brain barrier?

130. If I was pre-menopausal before undergoing chemotherapy and therefore ineligible to take tamoxifen, and I am now post-menopausal after having undergone chemotherapy, am I now eligible to take tamoxifen as a preventative of recurrence?

131. Can I have an oophorectomy instead of having chemotherapy?

Appendix C

Tutorial for Category Tool

Introduction

The category tool tries to group documents into categories that answer the question used to find the documents. To find a document about a particular topic, look for a category that matches that topic or a more general topic. The categories are arranged in a hierarchy such that the more specific categories appear indented under the more general category. Each document may appear in multiple categories, and specific categories may be listed under multiple general categories.

Summary of Screens

- Top part of computer screen:
  - shows the question asked
  - shows the number of documents (or references) returned from a search of cancer articles
- Left part of computer screen:
  - shows the top two levels of document categories (more specific categories are shown indented under their more general category)

- shows in parentheses the number of documents or references that belong to each category
- allows you to click on the underlined number of references for a category, which brings that category and the titles of its documents to the top of the right part of the screen

• Right part of screen can show one of two things:

- a hierarchical list of categories with the list of the titles that belong to a category shown below it (titles and categories may appear more than once in this section)
- a document, including its title , identification number, author, journal, keywords, and abstract

Changing What is Displayed

- Scroll up and down either the left or the right screen (if it has a scroll bar at its rightmost edge) by clicking on the arrows in the scroll bar or by clicking above or below the highlight section in the scroll bar.
- Display a category, its subcategories, and corresponding document titles by clicking on the number of references underlined in the left screen. It will appear at the top of the right screen.
- Display the title, identification number, author, journal, keywords, and abstract for a document by clicking on its underlined title in the right screen. It will appear in the right screen.
- Bring back the entire list of categories and documents within that category by clicking on the underlined number of references in parentheses after the category name on the left screen. It will appear at the top in the right screen.

Try Using the Tool

- Bring the group of documents that are in the diet category to the top of the right screen.
- Display the abstract of a document in the diet category.
- Bring the group of documents that are about population characteristics to the top of the right part of the screen.

- Scroll to the bottom of the right screen to find the title of the document that appears last in the list.
- Find the identification number of a document that discusses oral contraceptives as a risk factor.

Appendix D

Tutorial for Cluster Tool

Introduction

The cluster tool tries to group documents that discuss similar topics into clusters. Each cluster is labeled by words that are the most representative of that cluster of documents. To find a document about a particular topic, look for a cluster label that matches the topic, part of the topic, or a more general topic. Each document will appear in only one cluster.

Summary of Screens

- Top part of computer screen:
  - shows the question asked
  - shows the number of documents (or references) returned from a search of cancer articles
- Left part of computer screen:
  - shows each document cluster with the words that describe that cluster appearing below the cluster number
  - shows in parentheses the number of documents or references that belong to each cluster

- allows you to click on the underlined number of references for a cluster, which brings that cluster and the titles of its documents to the top of the right part of the screen

- Right part of screen can show one of two things:
- a list of clusters with the list of the titles that belong to a cluster shown below it
- a document, including its title, identification number, author, journal, keywords, and abstract

Changing What is Displayed

- Scroll up and down either the left or the right screen (if it has a scroll bar at its rightmost edge) by clicking on the arrows in the scroll bar or by clicking above or below the highlight section in the scroll bar.
- Display a cluster and its corresponding documents by clicking on the number of references or documents that is underlined in parentheses on the left screen. It will appear at the top of the right screen.
- Display the title, identification number, author, journal, keywords, and abstract for a document by clicking on its underlined title in the right screen. It will appear in the right screen.
- Bring back the entire list of clusters and documents within that cluster by clicking on the underlined number of references in parentheses after the category name on the left screen. It will appear at the top in the right screen.

Try Using the Tool

- Bring the group of documents that are in the cluster described by the words epidemiology and years to the top of the right screen.
- Display the abstract of any document in the cluster described by the words prevention and years.
- Bring the group of documents that are about surgery to the top of the right part of the screen.
- Scroll to the bottom of the right screen to find the title of the document that appears last in the list.

- Find the identification number of a document that discusses family history as a risk factor.

Appendix E

Tutorial for Ranking Tool

Introduction

The ranking tool tries to rank documents according to how relevant they are to the question. They are ranked from most relevant (the first document) to least relevant (the last document).

Summary of Screens

- Top part of computer screen:
  - shows the question asked
  - shows the number of documents (or references) returned from a search of cancer articles
- Left part of computer screen:
  - shows the groups of ranked documents, in groups of ten
  - allows you to click on the underlined ranking of the group which brings that group to the top of the right part of the screen
- Right part of screen can show one of two things:

- the list of the titles of all documents (or references) that were returned from the search
- a document, including its title, identification number, author, journal, keywords, and abstract

Changing What is Displayed

- Scroll up and down either the left or the right screen (if it has a scroll bar at its rightmost edge) by clicking on the arrows in the scroll bar or by clicking above or below the highlight section in the scroll bar.
- Display a group of ranked documents by clicking on the range of ranked documents underlined in the left screen. It will appear at the top of the right screen.
- Display the title, identification number, author, journal, keywords, and abstract for a document by clicking on its underlined title in the right screen. It will appear in the right screen.
- Bring back the entire list of ranked documents with a range of documents by clicking on the underlined range of documents on the left screen. It will appear at the top in the right screen.

Try Using the Tool

- Bring the group of documents that are ranked 31-40 to the top of the right part of the screen
- Display the abstract of the document ranked 27.
- Bring the group of documents that are ranked 21-30 to the top of the right part of the screen
- Find the title of the document that is ranked last in the entire list.
- Find the identification number of a document that discusses obesity as a risk factor.

Appendix F

Timed Tasks for Each Query

Query: "What are the ways to prevent breast cancer?

1) In the next four minutes, list as many methods for preventing cancer as you can. They must be discussed in these documents but you must not have listed them originally:

2) Can hormone therapy be used in breast cancer prevention? Write down the answer given in one of the documents, and write down that document's identification number:

3) Can diet be used in the prevention of breast cancer? Write down the answer given in one of the documents, and write down that document's identification number:

Query: "What are the prognostic factors for breast cancer?"

1) In the next four minutes, list as many factors that influence breast cancer prognosis as you can. They must be discussed in these documents but you must not have listed them originally:

2) Does the extent of lymphatic invasion influence prognosis? Write down the answer given in one of the documents, and write down that document's identification number.

3) Can someone have a recurrence after she has had a mastectomy? Write down the answer given in one of the documents, and write down that document's identification number.

"What are the treatments for breast cancer?"

1) In the next four minutes, list as many treatments for breast cancer as you can. They must be discussed in these documents but you must not have listed them originally:

2) For patients with stage I or stage II breast cancer, is a mastectomy or lumpectomy recommended? Write down the answer given in one of the documents, and write down that document's identification number.

3) Should someone have radiation therapy after a lumpectomy? Write down the answer given in one of the documents, and write down that document's identification number.

Appendix G

Instructions for Organizing Documents

Instructions

1) Read through the Categories provided. They are organized into a hierarchy where more specific categories are indented under the more general categories. You will be using only the categories that are prefaced by an A and number in parentheses. The more general categories are only there to make it easier for you to find the more specific categories. Note that some of the specific categories appear under more than one general category. They all have the same number associated with them, so you only need to assign it once.

2) Read through the Clusters provided. Each cluster (C1-C5)[1] is described by the commonly occurring words from that cluster. The vertical bars separate the words to indicate that each word should be considered individually, not part of a phrase.

3) For each citation provided:

1. The number of clusters provided on the instructions varied from 3 to 5, corresponding to the query that was used.

* Read the title and abstract, carefully thinking about how the citation answers the original query: "What are the prognostic indicators for breast cancer?".

* Assign and write on the citation as many categories (A1-A69)[1] as appropriate that both describe that citation and answer the query. If you do not think the document is relevant to answering the query, write NR on the citation and do not write any other category from the A group. If you think an additional category is necessary for describing the citation, write OC on the citation and provide the category label that you think is appropriate. You may assign the category OC in addition to other categories. Make sure you only assign categories that make sense for answering the query.

* Assign the one cluster (C1-C5) that most closely matches that citation. Assign a cluster even if you have entered NR for the category. The categories and the clusters are two different ways of grouping the citations. There is no relationship between the clusters (C1-C5) and the categories (A1-A69).

4) When you have finished assigning categories and clusters to all of the citations, please answer the questions provided on the back of this form:

Using the scale below, please answer the following questions:

1 = Almost never
2 = Some of the time
3 = Almost half of the time
4 = Most of the time
5 = Almost always

For the Categories: (A1-A69):

1) The labels on the categories are meaningful.

1 2 3 4 5

1. The number of categories in the instructions varied from 27 to 71, corresponding to the query that was used.

2) The categories correspond to groups of documents that are appropriate for the citations provided.

1 2 3 4 5

3) The categories correspond to groups of documents that are appropriate for the original query, "What are the prognostic indicators for breast cancer?"[1]

1 2 3 4 5

For the Clusters: (C1-C5):

4) The labels on the clusters are meaningful.

1 2 3 4 5

5) The clusters correspond to groups of documents that are appropriate for the citations provided.

1 2 3 4 5

6) The clusters correspond to groups of documents that are appropriate for the original query, "What are the prognostic indicators for breast cancer?"

1 2 3 4 5

1. The query in the instructions corresponded to one of the three to which that the subject was assigned.

A p p e n d i x H

View of Categories in Evaluation of Technical Claim

Categories corresponding to query:

"What are the diagnostic tests for breast cancer?"

- Chemicals and Drugs
  - **(A1)** Antigens, Tumor-Associated, Carbohydrate
  - **(A2)** Protein p53
  - **(A3)** Receptors, Epidermal Growth Factor-Urogastrone
  - **(A4)** Receptors, Estrogen
  - **(A5)** Receptors, Progesterone
  - **(A6)** Tumor Markers, Biological
- Diagnosis
  - **Diagnostic Errors**
    - **(A7)** False Negative Reactions
    - **(A8)** False Positive Reactions
  - **Diagnostic Techniques and Procedures**
    - **(A9)** Diagnostic Imaging
      - **(A10)** Magnetic Resonance Imaging

        - **(A11)** Mammography
        - **(A12)** Radiographic Image Enhancement
        - **(A13)** Spectroscopy, Near-Infrared
        - **(A14)** Subtraction Technique
        - **(A15)** Tomography
        - **(A16)** Ultrasonography, Mammary
      - **Diagnostic Techniques, Surgical**
        - **(A17)** Biopsy
        - **(A18)** Biopsy, Needle
      - **(A19)** Mass Screening
      - **(A20)** Medical History Taking
      - **(A21)** Neoplasm Staging
      - **(A22)** Physical Examination
        - **(A23)** Breast Self-Examination
        - **(A24)** Palpation
    - **Laboratory Techniques and Procedures**
      - **(A17)** Biopsy
      - **(A18)** Biopsy, Needle
- Investigative Techniques
    - **(A25)** Flow Cytometry
    - **(A26)** Immunohistochemistry
    - **(A19)** Mass Screening
    - **(A13)** Spectroscopy, Near-Infrared
    - **(A27)** Spectrum Analysis, Raman
- (NR) Not Relevant to the query
- (OC) Other Category -- Please specify a category name

A p p e n d i x I

View of Clusters in Evaluation of Technical Claim

Clusters corresponding to query:
"What are the diagnostic tests for breast cancer?"

(C1) cancer | women | screening | age | patients | clinical | years | positive | mammography | factors | survival | results

(C2) carcinoma | pathology | surgery | imaging | disease | cancer | patients | biopsy | diagnostic | surgical | situ | refs

(C3) imaging | lesions | contrast | mri | images | biopsy | mammography | methods | needle | enhanced | benign | results

Bibliography

ACM (1997). 1991 ACM Computing Classification System. [Online] Available at http://www.acm.org/class/1991/.

Allan J, Hirsch M (1997). A Graphic Interface for User Directed Clustering of Retrieved Documents (abstract). *American Medical Informatics Association (AMIA) Spring Symposium*, San Jose, CA.

Allen RB, Obry P, Littman M (1993). An interface for navigating clustered document sets returned by queries. *ACM SIGOIS: Conference on Organizational Computing Systems (COOCS)*, Milpitas, CA:166-171.

AltaVista (1997). AltaVista Technology, Inc. [Online] Available at http://www.altavista.com/.

Altman RB, Bada M, Chai XJ, Chen RO, Abernethy NF (1999). Using Ontologies for a collaborative scientific data resource in molecular biology: The RIBOWEB System. *IEEE Intelligent Systems, Special Issue on Ontologies*. (in press)

Apte C, Damerau F, Weiss SM (1994). Automated Learning of Decision Rules for Text Categorization. *Transactions of Office Information Systems*; 12(3).

Bada MA, Altman RB (1999). Computational Modeling of Structured Experimental Data. *Methods in Enzymology.* (in press)

Baldonado M, Winograd T (1997). SenseMaker: An Information-Exploration Interface Supporting the Contextural Evolution of a User's Interests. *Computer Human Interaction (CHI)*, Atlanta.

Baldonado MQW (1997). Searching, browsing, and metasearching with SenseMaker. *WEB Techniques*; 2(5):Inclusive Pagination: p. 42-47.

Belkin NJ, Croft WB (1987). Retrieval techniques. In: Williams ME, editor, ed. *Annual review of information science and technology.* Amsterdam, Netherlands: Elsevier; p. 109-145.

Bernstein LM, Williamson RE (1984). Testing of a Natural Language Retrieval System for a Full Text Knowledge Base. *Journal of the American Society for Information Science (JASIS)* 35(4): 235-247.

Borgman CL (1986). Why are online catalogs hard to use? Lessons learned from information-retrieval studies. *Journal of the American Society for Information Science*; 37(6):387-400.

Buckley C, Salton G, Allan J (1994). The effect of adding relevance information in a relevance feedback environment. *SIGIR '94. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Berlin, Germany, Springer-Verlag.

Card SK, Robertson GG, York W (1996). The webbook and the web forager: An information workspace for the world-wide web. *ACM SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, Canada.

Carpineto C, Romano G (1995). ULYSSES: a lattice-based multiple interaction strategy retrieval interface. *Human-Computer Interaction. 5th International Conference, EWHCI'95. Selected Papers*, Berlin, Germany:p. 91-104.

CBHP (1997). Welcome to the Community Breast Health Project. [Online] Available at http://www-med.stanford.edu/CBHP/.

Cheeseman P, Kelly J, Self M, Stutz J, Taylor W, Freeman D (1988). AutoClass: a Bayesian Classification System. *Proceedings of the Fifth International Conference on Machine Learning*, :54-64.

CIIR (1997). Natural Language Processing Laboratory, University of Massachusetts. [Online] Available at http://www-nlp.cs.umass.edu/~nlpgroup/nlpie.html.

Cimino JJ, Aguirre A, Johnson SB, Peng P (1993). Generic queries for meeting clinical information needs. *Bulletin of the Medical Library Association* 81(2): 195-206.

Cohen J (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1): 37-46.

Crowder RG (1976). *Principles of Learning and Memory.* Hillsdale, NJ: Erlbaum.

Cutting D, Karger D, Pedersen J, Tukey JW (1992). Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections. *SIGIR '92. Proceedings of the Fifteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, .

Cycorp (1997). Welcome to the Cyc Public Ontology. [Online] Available at http://www.cyc.com/public.html.

Deerwester S, Dumais S, Furnas G, Landauer T, Harshman R (1990). Indexing by Latent Semantic Analysis. *Jorunal of the American Society for Information Science (JASIS)*; 41(6):391-407.

Doll W, Torkzadeh F (1988). The measurement of end-user computing satisfaction. *MIS Quarterly* 12: p. 259-274.

Dumais ST (1993). Latent Semantic Indexing (LSI) and TREC-2. *The Second Text REtrieval Conference (TREC-2)*, :105-115.

Efthimiadis EN (1993). A user-centred evaluation of ranking algorithms for interactive query expansion. *SIGIR '93. Proceedings of the Sixteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*.

Egan DE, Remde JR, Gomez LM, Landauer TK, Eberhardt J, Lochbaum CC (1989). Formative design-evaluation of SuperBook. *ACM Transactions on Information Systems* 1: p. 30-57.

Ellis D, Furner-Hines J, Willett P (1994). On the measurement of inter-linker consistency and retrieval effectiveness in hypertext databases. *SIGIR '94. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Berlin, Germany, Springer-Verlag.

Evans M (1993). Structured abstracts: rationale and construction. *European Journal of Surgery*; 159(3):131-132.

Fellbaum C, Ed. (1998). *WordNet: An Electronic Lexical Database*, MIT Press.

Funk ME, Reid CA (1983). Indexing consistency in MEDLINE. *Bulletin of the Medical Library Association*; 71(2):176-183.

Glantz SA (1997). *Primer of Biostatistics*, McGraw-Hill.

Griffiths A, Luckhurst HC, Willett P (1986). Using inter-document similarity information in document retrieval systems. *Journal of the American Society for Information Science*(37):3-11.

Grishman R, Sundheim B (1996). Message Understanding Conference - 6: A Brief History. *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark.

Guha RV, Lenat DB (1994). Enabling Agents to Work Together. *Communications of the ACM*; 37(7).

Harman D (1992). Ranking Algorithms. *Information Retrieval Data Structures & Algorithms*. R. B.-Y. William B. Frakes, Prentice Hall.

Haynes R, Wilczynski N, McKibbon K, Walker C, Sinclair J (1994). Developing optimal search strategies for detecting clinically sound studies in MEDLINE. *Journal of the American Medical Informatics Association* 1(6): p. 447-458.

Hearst M. (to appear). Categories, Clusters, and Attributes. In: Strzalkowski, ed. *Natural Language Information Retrieval*: Kluwer Academic Publishers.

Hearst M, Karadi C (1997). Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy. *SIGIR '97: Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, PA.

Hearst MA, Pedersen JO (1996). Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. *SIGIR '96: Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.*

Hearst MA (1995). TileBars: visualization of term distribution information in full text information access. *Human Factors in Computing Systems. CHI'95 Conference Proceedings*, New York, NY, USA:p. 59-66.

Hearst MA, Karger DR, Pedersen JO (1995). Scatter/Gather as a tool for the navigation of retrieval results. *AI Applications in Knowledge Navigation and Retrieval. Papers from the 1995 AAAI Fall Symposium (Tech. Report FS-95-03)*, Menlo Park, CA, USA: p. 65-71.

Heller MB (1991). Structured abstracts: a modest dissent. *Journal of Clinical Epidemiology*; 44(8):739-740.

Hersh WR, Greenes RA (1989). SAPHIRE – an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval and hierarchical relationships. *Computers and Biomedical Research*; 23:410-425.

Hersh WR, Greenes RA (1990). Information retrieval in medicine: state of the art. *M.D. Computing*; 7(5):302-311.

Hersh WR, Hickam DH (1992). A comparison of retrieval effectiveness for three methods of indexing medical literature. *The American Journal of the Medical Sciences*; 303(5):292-300.

Hersh WR, Hickam DH (1993). A comparison of two methods for indexing and retrieval from a full-text medical database. *Medical Decision Making*; 13(3):220-226.

Hull D (1994). Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing. *Proceedings of the 17th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, :282-289.

Huth EJ (1987). Structured abstracts for papers reporting clinical trials. *Annals of Internal Medicine*; 106(4):626-627.

Humphrey SM (1992). Indexing biomedical documents: from thesaural to knowledge-based retrieval systems. *Artificial Intelligence in Medicine* 4: 343-71.

Humphreys BL, Lindberg DA, Schoolman HM, Barnett GO (1998). The Unified Medical Language System: an informatics research collaboration. *Journal of the American Medical Informatics Association* 5(1): 1-11.

Jardine N, van Rijsbergen CJ (1971). The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval*(7):217-240.

Keen EM (1991). The use of term position devices in ranked output experiments. *Journal of Documentation* 47: p. 1-22.

Koller D, Sahami M (1996). Toward Optimal Feature Selection. *In ICML-96: Proceedings of the Thirteenth International Conference on Machine Learning*, San Francisco, CA:284-292.

Krishnaiah PR, Kanal LN (1982). *Classification, Pattern Recognition, and Reduction in Dimensionality.* Amsterdam.

Kwok KL (1996). A New Method of Weighting Query Terms for Ad-Hoc Retrieval. *SIGIR '96: Proceedings of the Nineteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland.

Landis JR, Koch GG (1977). The measurement of observer agreement for categorical data. *Biometrics*; 33:159-174.

Lenat DB (1995). CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*; 38(11).

Lewis D, Ringuette M (1994). A Comparison of Two Learning Algorithms for Text Categorization. *Symposium on Document Analysis and Information Retrieval*, University of Nevada, Las Vegas.

Lewis DD (1992a). Representation and Learning in Information Retrieval. PhD Thesis. Department of Computer Science, University of Massachusetts, Amherst.Kleiboemer AJ, Lazear MB, Pederson JO (1996). Tailoring a retrieval system for naive users. *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*, Las Vegas, NV.

Lewis DD (1992b). An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. *Proceedings of the 15th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, :37-50.

Lilleyman J, Lowe D (1992). Structured abstracts. *Journal of Clinical Pathology*; 45(1):8.

Lock S (1988). Structured abstracts. *The British Medical Journal*; 297(156):6642.

Lycos (1997). Lycos Pro Search. [Online] Available at http://www.lycospro.lycos.com.

Massand B, Linoff G, Waltz D (1992). Classifying news stories using memory-based reasoning. *Proceedings of the 15th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, :59-65.

McCray AT, Aronson AR, Browne AC, Rindflesch TC, Razi A, Srinivasan S (1993). UMLS knowledge for biomedical language processing. *Bulletin of the Medical Library Association*; 81(2):184-194.

Mezzich JE, Kraemer HC, Worthington DRL, Coffman GA (1981). Assessment of Agreement Among Several Raters Formulating Multiple Diagnoses. *Journal of Psychiatric Research* 16(1): p. 29-39.

Miller GA (1995). WordNet: A Lexical database for English. *Communications of the ACM*; 38(11):39-41.

Miller GA (1997). WordNet. [Online] Available at http://www.cogsci.princeton.edu/~wn/.

MIT (1997). Common Lisp Hypermedia Server (CL-HTTP). [Online] Available at http://wilson.ai.mit.edu/cl-http/frame.html.

MUC-3 (1991). *Proceedings of the 3rd Message Understanding Conference (MUC-3)*: Morgan Kauffman Publishers, Inc.

MUC-4 (1992). *Proceedings of the 3rd Message Understanding Conference (MUC-4)*: Morgan Kauffman Publishers, Inc.

MUC-5 (1993). *Proceedings of the 5th Message Understanding Conference (MUC-5)*: Morgan Kauffman Publishers, Inc.

MUC-6 (1995). *Proceedings of the 6th Message Understanding Conference (MUC-6)*. Columbia, MD: Morgan Kauffman Publishers, Inc.

Nelson SJ, Cole WG, Tuttle MS, Olson NE, Sherertz DD (1994). Recognizing new medical knowledge computationally. *Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC). Patient-Centered Computing*, New York, NY, USA:p. 409-413.

Ng HT, Goh WB, Low KL (1997). Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization, Proceedings of the 20th International ACM/SIGIR Conference on Research and Development in Information Retrieval:67-73.

NLM (1997a). Medical Subject Headings Fact Sheet. [Online] Available at http://www.nlm.nih.gov/pubs/factsheets/mesh.html.

NLM (1997b). NLM Online Databases and Databanks. [Online] Available at http://wwwindex.nlm.nih.gov/pubs/factsheets/online_databases.html.

NLM (1997c). Welcome to PubMed. [Online] Available at http://www.ncbi.nlm.nih.gov/PubMed/

NCI (1998). CancerNet. [Online] Available at http://cancernet.nci.nih.gov/.

NLM (1998a). NLM Online Databases and Databanks. [Online] Available at http://wwwindex.nlm.nih.gov/pubs/factsheets/online_databases.html.

NLM (1998b). PubMed Clinical Queries. [Online] Available at http://www.ncbi.nlm.nih.gov/PubMed/clinical.html.

NLM (1998c). The UMLS Metathesaurus Fact Sheet. [Online] Available at http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html.

NLM (1999a). Medical Subject Headings Fact Sheet. [Online] Available at http://www.nlm.nih.gov/pubs/factsheets/mesh.html.

NLM (1999b). The UMLS Metathesaurus Fact Sheet. [Online] Available at http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html.

NLM (1999c). UMLS Semantic Network Fact Sheet. [Online] Available at http://www.nlm.nih.gov/factsheets/umlssemn.html.

Pirolli P, Schank P, Hearst MA, Diehl C (1996). Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection. *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*.Purcell GP (1996). Contextual document models for searching the clinical literature. PhD Thesis. Section on Medical Informatics, Stanford University.

Purcell GP (1996). Contextual document models for searching the clinical literature. Ph.D. thesis. Section on Medical Informatics, Stanford University, Stanford.

Purcell GP, Rennels GD, Shortliffe EH (1997). Development and Evaluation of a Context-Based Document Representation for Searching the Medical Literature. *International Journal on Digital Libraries*; 1(3):p. 288-296.

Rasmussen E (1992). Clustering Algorithms. In: William B. Frakes RB-Y, ed. *Information Retrieval Data Structures & Algorithms*: Prentice Hall; 419-442.

Rennels GD (1987). A computational model of reasoning from the clinical literature. In: Reichertz PL, Lindberg DAB, eds. *Lecture Notes in Medical Informatics*. Berlin: Springer-Verlag; 230.

Review M (1997). Mathematical Review 1991 Subject Classification. [Online] Available at http://www.ma.hw.ac.uk/~chris/MR/MR.html.

Riloff E (1993). Automatically constructing a dictionary for information extraction tasks. *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Menlo Park, CA, USA, AAAI Press.

Riloff E (1996a). An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence* 1-2: p. 101-34.

Riloff E (1996b). Using learned extraction patterns for text classification. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*. S. Wermter, editor, E. Riloff, editor and G. Scheler, editor. Berlin, Germany, Springer-Verlag**:** p. 275-89.

Robertson GC, Card SK, MacKinlay JD (1993). Information visualization using 3D interactive animation. *Communications of the ACM* 36: p. 56-71.

Robertson SE, Walker S (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *SIGIR '94. Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Berlin, Germany, Springer-Verlag.

Robertson SE, Walker S, Jones S, Hancock-Beaulieu MM, Gatford M (1994). Okapi at TREC-2. *The Second Text Retrieal Conference (TREC-2)*, Gaithersburg, MD.

Rowley JE (1996). *Organizing Knowledge: an introduction to information retrieval*. Aldershot, England, Gower.

Sahami M, Yusufali S, Baldonado MQW (1997a). Real-time Full-text Clustering of Networked Documents (Abstract). *AAAI-97: Proceedings of the Fourteenth National Conference on Artificial Intelligence*., Providence, RI:845.

Sahami M (1998). Using Machine Learning to Improve Information Access. PhD Thesis. Computer Science Department. Stanford, Stanford University.

Sahami M, Yusufali S, Baldonado MQW (1998). SONIA: A Service for Organizing Network Information Autonomously. *Digital Libraries 98: Proceedings of the Third ACM Conference on Digital Libraries*, Pittsburgh, PA, USA.

Salton G, Wong A, Yang CS (1975). A vector space model for automatic indexing. *Communications of the ACM* 18: 613-620.

Salton G, Yang CS, Yu CT (1975). A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science* 26: 33-44.

Salton G, McGill MJ (1983). *Introduction to modern information retrieval*. New York, McGraw-Hill, Inc.

Salton G, Buckley C (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*; 24(5):513-523.

Salton G (1989). *Automatic Text Processing*: Addison-Wesley.

Salton G (1989). *Automatic Text Processing*, Addison-Wesley.Sim I, Rennels G (1995). A Trial Bank Model for the Publication of Clinical Trials. *Nineteenth Annual Symposium on Computer Applications in Medical Care*, New Orleans, LA:863-867.

Sim I (1997). Trial Banks: An informatics foundation for evidence-based medicine. PhD Thesis. Medical Information Sciences, Stanford University, Stanford.

Small H, Sweeney E (1985). Clustering the Science Citation Index using cocitations. I. A comparison of methods. *Scientometrics*; 7:p. 391-409.

Soderland S, CRYSTAL (1996): Learning Domain-specific Text Analysis Rules. University of Massachusetts, Center for Intelligent Information Retrieval (CIIR).

Soderland S, Fisher D, Aseltine J, Lehnert W (1995). CRYSTAL: Inducing a Conceptual Dictionary. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, :pp. 1314-1319.

Squires BP, Keith RG, Meakins JL (1992). Structured abstracts for clinical research manuscripts and reviews. *Canadian Journal of Surgery*; 35(5):473-475.

Tong RM, Appelbaum LA (1994). Machine learning for knowledge-based document routing (a report on the TREC-2 experiment). *Second Text REtrieval Conference (TREC-2) (NIST-SP 500-215)*, Washington, DC, USA:p. 253-264.

Tuttle MS, Nelson SJ (1994). The role of the UMLS in 'storing' and 'sharing' across systems. *International Journal of Bio-Medical Computing*; 1-4:p. 207-237.

Tuttle MS, Sherertz DD, et al. (1994). Toward an interim standard for patient-centered knowledge-access. *Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC). Patient-Centered Computing*, New York, NY, USA, McGraw-Hill.

van Rijsbergen CJ, Croft WB (1975). Document Clustering: An Evaluation of Some Experiments with the Cranfield 1400 collection. *Information Processing & Management*(11):171-182.

van Rijsbergen CJ (1971). *Information Retrieval*. London: Butterworths.

Voorhees EM (1985). The cluster hypothesis revisited. *Proceedings of ACM/SIGIR*, :p. 188-196.

Wade SJ, Willett P, Bawden D (1989). SIBRIS: the Sandwich Interactive Browsing and Ranking Information System. *Journal of the American Society for Information Science (JASIS)* 15: 249-260.

Warren KS (1981). *Coping with the biomedical literature*. New York: Praeger Publishers.

Wiener E, Pedersen J, Weigend AS (1995). A Neural Network Approach to Topic Spotting. *Fourth Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV.

Willett P (1988). Recent trends in hierarchic document clustering: a critical review. *Information Processing & Management*; 24(5):p. 577-597.

Yahoo! (1997). Yahoo! [Online] Available at http://www.yahoo.com/.

Yang Y, Chute CG (1994a). An application of expert network to clinical classification and MEDLINE indexing. *Eighteenth Annual Symposium on Computer Applications in Medical Care (SCAMC)*, Washington, DC:p. 157-161.

Yang Y, Chute CG (1994b). An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*; 12(3):p. 252-277.