A Knowledge-Based Approach to Organizing Retrieved Documents

Wanda Pratt
Information & Computer Science
University of California, Irvine
Irvine, CA 92697-3425
pratt@ics.uci.edu

Marti A. Hearst
School of Information Management & Systems
UC Berkeley, 102 South Hall
Berkeley, CA 94720-4600
hearst@sims.berkeley.edu

Lawrence M. Fagan
Stanford Medical Informatics
Stanford University
Stanford, CA 94305-5479
fagan@smi.stanford.edu

Abstract

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. In this paper, we present DynaCat, a tool that dynamically categorizes search results into a hierarchical organization by using knowledge of important kinds of queries and a model of the domain terminology. Results from our evaluation show that DynaCat helps users find answers to those important types of questions more quickly and easily than when they use a relevance-ranking system or a clustering system.

Introduction

Current information-retrieval tools usually return results that consist of a simple list of documents. Such long, undifferentiated lists can overwhelm people and cause them to abandon their search before they assess the available information.

Most search tools assist in solving the problem of too many search results by helping the user reformulate her query into a more specific one. However, even if a user 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 be confronted with a very long list of documents, and would need tools to help her understand those documents. By focusing on query formulation, the search-tool developers assume that relevant documents are few. However, 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 provides a more specific query, she may miss valuable, relevant documents.

We have created an approach that addresses this problem by dynamically categorizing search results into meaningful groups that correspond to the user's query. Our approach uses knowledge of important kinds of queries and a model of the domain terminology to create the hierarchical categorization of search results. We have implemented this approach in a tool called **DynaCat** 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 1998). Our approach summarizes the information returned from a search by placing the retrieved documents into useful categories, thus helping users to gain quick and easy access to important information.

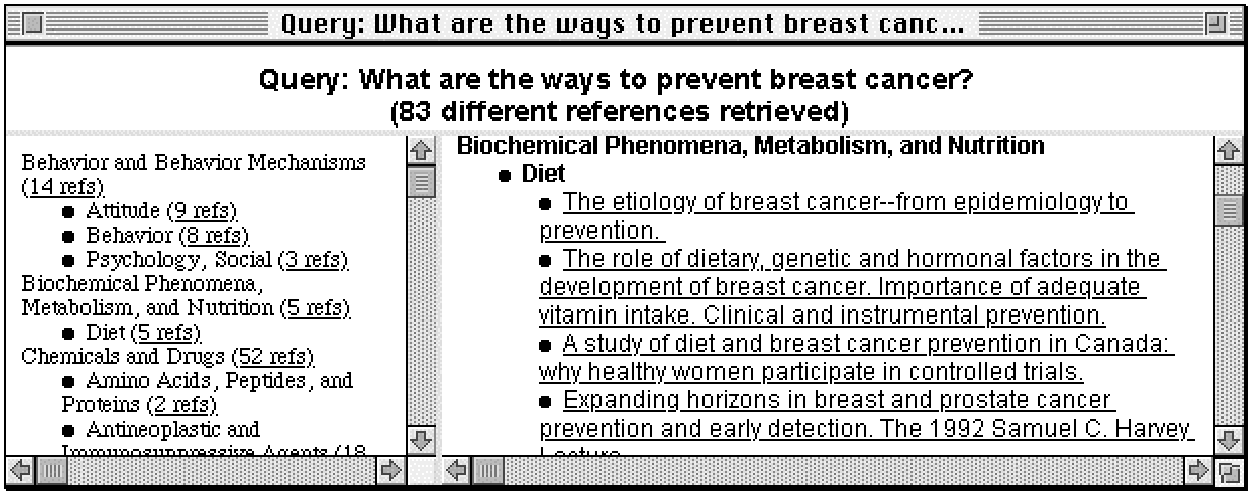
Search Scenario

The amount of medical literature continues to grow and specialize. At the same time, many patients and their families are becoming more proactive in searching the medical literature for information regarding their medical problems, despite the fact that medical journal articles can be intimidating to read for lay people.

Consider a woman whose mother was diagnosed recently with breast cancer. She is worried about her own chances of developing breast cancer and 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 available in medical journal articles.

She could choose to search the primary medical literature using PubMed, the free, web-based MEDLINE search tool. If she searches for documents in the previous year that use the keywords *breast neoplasms* (a more general medical term for breast cancer) and *prevention*, PubMed returns the titles of over 400 documents displayed as a long 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, she could see the search results organized by the preventive actions found in those documents. Figure 1 shows the interface


Query: What are the ways to prevent breast canc...

Query: What are the ways to prevent breast cancer?
(83 different references retrieved)

Behavior and Behavior Mechanisms (14 refs)
- Attitude (9 refs)
- Behavior (8 refs)
- Psychology, Social (3 refs)

Biochemical Phenomena, Metabolism, and Nutrition (5 refs)
- Diet (5 refs)

Chemicals and Drugs (52 refs)
- Amino Acids, Peptides, and Proteins (2 refs)
- Antineoplastic and

Biochemical Phenomena, Metabolism, and Nutrition
- **Diet**
  - The etiology of breast cancer--from epidemiology to prevention.
  - The role of dietary, genetic and hormonal factors in the development of breast cancer. Importance of adequate vitamin intake. Clinical and instrumental prevention.
  - A study of diet and breast cancer prevention in Canada: why healthy women participate in controlled trials.
  - Expanding horizons in breast and prostate cancer prevention and early detection. The 1992 Samuel C. Harvey

Figure 1. 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. The right pane displays all the categories in the hierarchy and the titles of the documents that belong in those categories.

generated by DynaCat for a search on the CancerLit database using the keywords *breast neoplasms* and *prevention*.

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.

Other Approaches

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

Relevance-ranking systems create an ordered list of search results (Harman 1992). The order of the documents is based on a measure of how likely it is that the document is relevant to the query. Even if the documents are ranked by relevance criteria, an ordered list does not give the user much information on the similarities or differences in the contents 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.

Document-clustering systems create groups of documents based on the associations among the documents (Allen, Obry et al. 1993; Hearst and Pedersen 1996) (Sahami, Yusufali et al. 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 usually do not use information about the user's query in forming the clusters. The document groups are labeled by words extracted from the clusters, usually chosen by an information-theoretic measure. Lists of words of this sort may be understandable if the contents of the cluster are cohesive, but a list of words is not as inviting to the general user as a well-selected category label.

A Knowledge-Based Method

To retrieve textual information, most information retrieval systems use statistical, word-based approaches. Knowledge-based techniques are seen as untenable because of the time and work required to create and maintain the necessary models for each domain. Our approach is domain-specific, but it takes advantage of an existing model for much of the knowledge, rather than requiring the developer to create and maintain a large, new model. It requires two types of models: a small query model that must be created by the developer, and a large domain-specific terminology model that should already exist. For the medical domain, DynaCat uses the terminology model created by the National Library of Medicine (NLM), the Unified Medical Language System (UMLS), which provides information on over 500,000 biomedical terms. Figure 2 illustrates how DynaCat extends the standard search process.

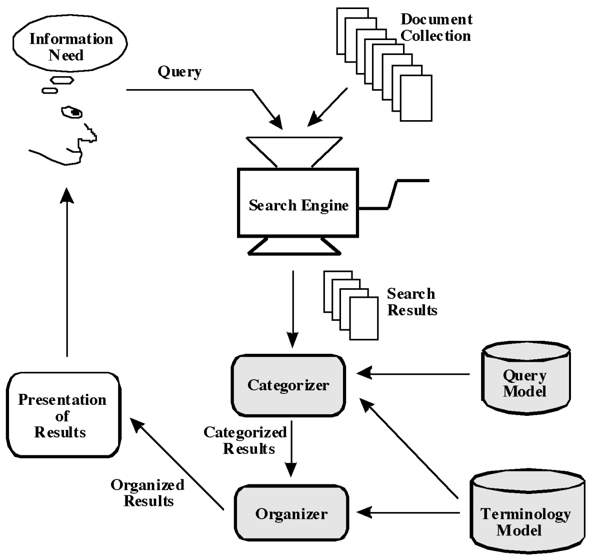


Figure 2. The search process using DynaCat. The components in light grey are the components that DynaCat adds to the traditional search process. These components do not influence which documents are retrieved, rather they determine how the search results are organized and displayed to the user.

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 how search results from those queries should be categorized. The **query model** provides this information through query types, and category 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. We 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?* 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 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 aspirin?* have the same *treatment—problems* query type, even though the questions refer to different treatments (e.g., the surgical procedure *mastectomy*, and the drug *aspirin*).

For DynaCat's medical query model (see Table 1), we created nine query types that correspond to questions that patients ask when they look for information in medical journal articles. We based this abstraction on a list of frequently-asked questions from a breast-cancer clinic. 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. Since the query model contains seven problem-oriented query types, the model covers at least 210,000 specific, problem-oriented queries.

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. 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, 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 we defined.

Table 1. 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 adverse effects of <treatment>?
Prognosis	
Problem—prognostic-indicators	What are the factors that influence the prognosis for <problem>?
problem—prognoses	What is the prognosis for <problem>?

For each query type, the system also needs an abstraction for the topics or categories that are appropriate for groups of search results. We call this abstraction **category types**. For example, when the user asks about the side effects of some drug, the types of categories that make

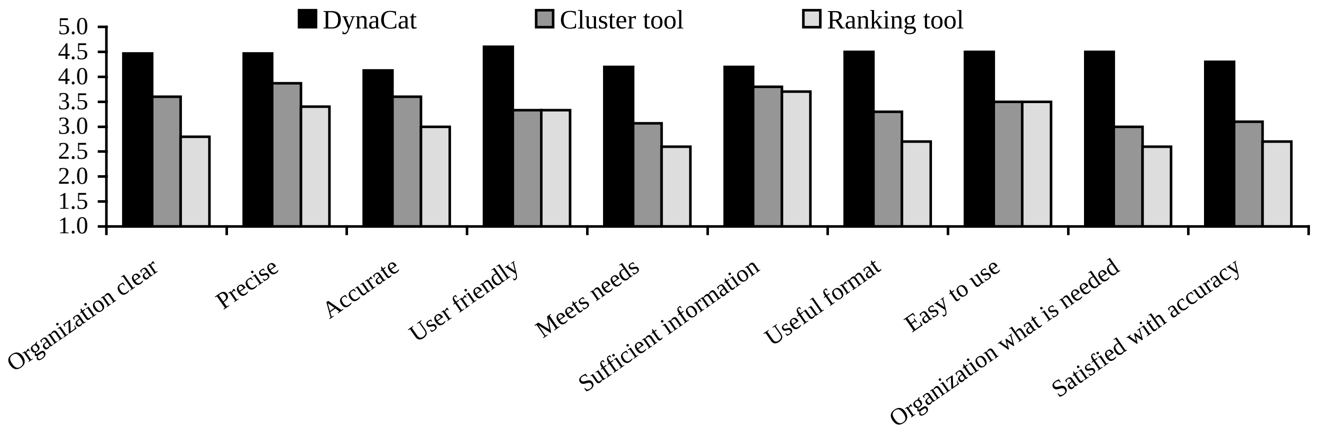


Figure 3. Results from the validated, user-satisfaction questionnaire. The mean values across all 15 subjects are shown on the y axis. The x axis shows a brief summary of the questions asked. Subjects answered the questions using a scale from 1 to 5, where 1 meant *almost never* and 5 meant *almost always* (the ideal answer). The difference between DynaCat and the cluster tool was statistically significant ($p < 0.05$) for all questions, as was that between DynaCat and the ranking tool, with the exception of the question about sufficient information where $p = 0.11$.

sense are those that indicate the various side 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*, and *prognostic-indicators*. As indicated by these names, each query type in the query model is linked to a category type, which determines the kinds of categories that DynaCat will generate 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 important, 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.

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 multiword phrases) to their corresponding general concept, called a **semantic type**. The UMLS medical terminology model links every term to at least one semantic type in a semantic network. For example, the term *penicillin* has a semantic type of *pharmacologic substance*. Individual, specific terms may become category labels if their semantic type is connected to the desired category type. 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*.

The query model does not contain references to specific terms in the terminology model; it only lists the appropriate semantic types for each category type. This representation isolates the query model from many changes that could occur in a terminology model. For example, if the NLM added a new drug to their terminology model and linked the new drug to its appropriate semantic type (*pharmacologic substance*), we would not need to modify the query model. If DynaCat were organizing the search results by the category type *treatments* and one of the retrieved documents uses the new drug as a keyword, that keyword would become a category label automatically.

Categorizer

Many published documents contain keywords that authors or indexers have selected to describe the documents' content. The **categorizer** takes advantage of this information to determine which categories to select and which documents to assign to those categories.

Because many of a document's keywords do not correspond to the user's query, the categorizer must prune the irrelevant keywords from the list of potential categories. To accomplish this task, the categorizer retrieves the categorization criteria for the corresponding query type, and examines each document in the set of results individually. 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. When a keyword satisfies all the categorization criteria, the categorizer adds the document to the category labeled with that keyword. If it has not created such a category already, it makes a new category labeled by that keyword. The categorizer checks every keyword in a document against the categorization criteria; thus, it may categorize each document under as many labels as is appropriate for the given query type.

Organizer

After a set of documents have been retrieved and assigned to categories, the categorizer then passes this information to the **organizer**, which arranges the categories into a hierarchy. 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 set thresholds. It 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 exceeds a predefined threshold, the categories are grouped under a more general label. DynaCat generates the more general label by traversing up the terminology model's hierarchy to find a term that is a parent to several document categories.

DynaCat is implemented in Common LISP, and makes use of the search engine developed by Lexical Technology, Inc. (Tuttle, et al. 1996). For more information on the system and algorithms see (Pratt 1997; Pratt 1999).

Evaluation

We performed a user study to assess the usefulness of the DynaCat approach. Our hypothesis is that the dynamic categorization of search results is more useful to users who have general questions than are the two other dominant dynamic approaches to organizing search results: ranked lists and clusters. We define a useful system as one that helps users to

- learn about the kinds of information that are available to answer their question
- find answers to their question efficiently and easily
- feel satisfied with their search experience

We recruited 15 breast-cancer patients and their family members for this evaluation. 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. For the clustering tool, we used SONIA (Sahami, et al. 1998). It creates clusters by using group-average hierarchical, agglomerative clustering to form the initial set of clusters, and then refines them with an iterative method. For the relevance ranking, we used a standard algorithm recommended by Salton for situations in which the queries are short and the vocabulary is technical (Salton and Buckley 1988). Each subject used all three different queries: *What are the ways to prevent breast cancer?* (83 documents retrieved), *What are the prognostic factors for breast cancer?* (81 documents retrieved), and *What are the treatments for breast cancer?* (78 documents retrieved), but we 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 the tools, we 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 learned was the number of new answers that the subject 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 more answers than they did with the other two tools. The mean number of answers found for DynaCat was 7.80, for the cluster tool was 4.53, and for ranking tool was 5.60. This difference was significant ($p < 0.05$).

For the second type of timed task, we 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. We 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.

We used a 26-question, user-satisfaction questionnaire to assess several aspects of the user's experience. The first 10 questions were adapted from a validated satisfaction questionnaire (Doll and Torkzadeh 1988). Figure 3 illustrates the results for that portion of the questionnaire. On all the questions that requested quantitative answers, the satisfaction scores for DynaCat were much higher than they were with either for the ranking tool or for the cluster tool. On 13 of the 14 questions, this difference was statistically significant ($p < 0.05$). For the yes—no questions, all 15 users affirmed that DynaCat makes sense, is helpful, is useful, and has clear labels. All 15 also said that they would use DynaCat again for another search. In comparison, for the cluster tool, the number of subjects that responded positively ranged between only 5 and 10. For the ranking tool, the number of positive responses ranged from 6 to 9. When asked if they were frustrated, one user said she was frustrated and another was somewhat frustrated when she used DynaCat. In contrast, 9 subjects were frustrated using the cluster tool, and 8 subjects were frustrated using the ranking tool. On the final subjective assessment, 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 of the number of answers that users found in a fixed amount of time 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, compared to only 47 percent for the cluster tool and only 60 percent for the ranking tool.

Conclusions and Contributions

In summary, we have presented a new, knowledge-based method for dynamically categorizing search results. We have explained how this method provides information about (1) what kinds of information are represented in (or are absent from) the search results, by hierarchically organizing the document categories and by providing meaningful labels for each category; (2) how the documents relate to the query, by making the categorization structure 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.

Although we created DynaCat exclusively for the domain of medicine, 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 this technique (Rowley 1996). For example, the Association for Computing Machinery uses a taxonomy of computing terms (Coulter, et al. 1998), and Mathematical Review sponsors a similar taxonomy of mathematics terminology (Review 1991).

We have demonstrated a successful application of AI technology for the field of medicine. Our evaluation suggests that DynaCat helps users find answers to certain types of questions more efficiently and easily than the common statistical approaches. Since the study involved a small number of queries, more evaluation is needed to allow for larger claims. Nevertheless, these initial results suggest that if we use knowledge about users' queries, and the kinds of organizations that are useful for those queries, we can provide users with helpful and satisfying search experiences. By providing a useful organization of medical search results, DynaCat can help lay people, patients and their families, explore the medical literature, and become informed about their own medical care. Health-care professionals who have used this system have also expressed enthusiasm for using it in their work.

Acknowledgements

This work was conducted with the support of NLM grant LM-07033 and NCI contract N44-CO-61025.

References

Allen, R. B.; Obry, P. and Littman, M. 1993. An interface for navigating clustered document sets returned by queries. In Proceedings of the ACM SIGOIS: Conference on Organizational Computing Systems (COOCS). Milpitas, CA, 166-171.

Cimino, J. J., et al. 1993. Generic queries for meeting clinical information needs. *Bulletin of the Medical Library Association.* 81(2): 195-206.

Coulter, N., French et al. 1998. Computing Classification System 1998: Current Status and Future Maintenance. Report of the CCS Update Committee, ACM Computing Reviews.

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

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

Haynes, R., et al. 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. A. and Pedersen, J. O. 1996. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In Proceedings of the ACM Conference on Research and Development in Information Retrieval: SIGIR '96, p. 76-84.

NLM. 1998. NLM Online Databases and Databanks. http://wwwindex.nlm.nih.gov/pubs/factsheets/online_databases.html.

Pratt, W. 1997. Dynamic organization of search results using the UMLS. In Proceedings of the American Medical Informatics Association (AMIA) Annual Fall Symposium, 480-4.

Pratt, W. 1999. Dynamic Categorization: A Method for Decreasing Information Overload. Medical Information Sciences. Stanford, Stanford University.

Review,. Mathematical Review 1991 Subject Classification. http://www.ma.hw.ac.uk/~chris/MR/MR.html.

Rowley, J. E. 1996. *Organizing Knowledge: an introduction to information retrieval.* Aldershot, England, Gower.

Sahami, M.; Yusufali, S. and Baldonado, M. Q. W. 1998. SONIA: A Service for Organizing Network Information Autonomously. In Proceedings of the Digital Libraries 98. Pittsburgh, PA, USA.

Salton, G. and Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management.* 24(5): 513-23.

Tuttle, M. S., et al. 1996. Toward reusable software components at the point of care. In Proceedings of the American Medical Informatics Association (AMIA) Annual Fall Symposium, 150-4.