Appendix A

BENCHMARK TESTING

A.1 Benchmark Testing Design

We first performed a benchmark testing of the two algorithms using 30 sample queries (generated by the experimenters). Each query consisted of terms of various degrees of specificity (e.g., Artificial Intelligence vs. Natural Language Processing) and different numbers of search terms. We tested 5 cases each for queries with 1 term, 2 terms, 3 terms, 4 terms, 5 terms, and 10 terms, a total of 30 cases. A few examples of the queries used, all in the computing area, were: (1-term: Natural Language Processing), (2-terms: Group Decision Support Systems, Collaboration), (3-terms: Systems Analysis and Design, Simulation and Modeling, Optimization), etc.

A.2 Benchmark Testing Results and Discussion

For each query, we selected terms from different knowledge sources, "P" for the Public KB, "A" for the ACM CRCS, and "L" for the LCSH, as shown in Table A.1. Some terms may have appeared in more than one knowledge source. The results shown in Table A.1 reveal the number of iterations, the computing times,

and the sources of knowledge for the query terms and the system-suggested terms. It should be note that the branch-and-bound algorithm performed serial iteration, while the Hopfield net performed parallel relaxation at each iteration. The reason for investigating the source of knowledge for system-suggested terms was to show the extent to which each algorithm branched out and utilized knowledge from other knowledge sources.

In response time, the branch-and-bound algorithm clearly performed better than the Hopfield net parallel activation. A MINITAB two-sample t-test (Ryan et al., 1985) showed that on average the neural net took 24.5 seconds (standard deviation, STDEV = 8.34) while the semantic net took 6.9 seconds (STDEV = 2.42). The difference was statistically significant (value of two-sample t-test, T = 11.10 and significance level, P = 0.0000). This was clearly because the branch-and-bound search performed only a fixed number of serial explorations, while the Hopfield net searched a much larger search space during the parallel activation process.

Despite the variation in the number of starting terms, the response times for both methods increased only slightly when the number of starting terms was increased. This finding is important, especially when considering complex, fuzzy queries which often contain many starting terms (a scenario in which searchers need the most help from the system). The reason for this small variation was that our branch-and-bound search decided a threshold based on the user's expected

case	no. of terms	query terms in (P,A,L)	suggested terms in NN:(P,A,L)/SN:(P,A,L)	no. of iterat. NN/SN	times (secs) NN/SN
1	1	(1,1,1)	(12,7,7)/(5,7,2)	18/12	21/11
2	1	(1,0,1)	(5,0,16)/(19,0,2)	15/21	14/8
3	1	(1,1,1)	(11,5,11)/(15,0,0)	14/16	18/10
4	1	(0,0,1)	(0,0,20)/(0,0,20)	11/20	10/11
5	1	(1,0,1)	(4,4,19)/(16,0,3)	17/20	26/10
6	2	(2,1,0)	(19,2,3)/(23,1,0)	21/23	18/6
7	2	(2,0,2)	(16,0,8)/(18,0,1)	19/23	22/8
8	2	(2,0,0)	(20,3,4)/(21,0,0)	20/23	24/5
9	2	(2,1,1)	(11,5,11)/(19,0,0)	15/23	16/4
10	2	(2,1,2)	(11,0,12)/(20,0,0)	27/22	29/4
11	3	(3,0,1)	(20,0,18)/(18,0,0)	19/22	31/5
12	3	(1,2,1)	(4,11,8)/(14,0,2)	22/17	34/6
13	3	(2,1,3)	(22,1,8)/(15,0,2)	18/19	29/6
14	3	(1,3,1)	(20,2,2)/(19,0,0)	16/22	23/8
15	3	(1,2,2)	(13,9,3)/(18,0,1)	9/21	10/4
16	4	(2,2,4)	(17,4,4)/(16,1,1)	17/20	11/6
17	4	(3,2,2)	(11,2,13)/(19,0,1)	19/23	31/5
18	4	(2,3,2)	(18,5,6)/(17,0,2)	24/21	33/4
19	4	(1,3,4)	(18,2,5)/(20,1,1)	19/24	32/7
20	4	(1,2,1)	(15,8,3)/(12,2,1)	18/22	6/7
21	5	(1,4,1)	(19,4,6)/(19,0,1)	16/24	27/3
22	5	(4,2,2)	(10,1,12)/(19,0,1)	15/19	27/4
23	5	(3,2,4)	(2,0,18)/(0,0,21)	11/21	23/9
24	5	(5,0,1)	(19,0,3)/(17,0,1)	23/17	33/9
25	5	(5,0,1)	(20,0,1)/(23,0,0)	12/23	30/12
26	10	(8,0,3)	(11,0,13)/(12,0,19)	17/39	34/6
27	10	(10,1,3)	(13,2,10)/(18,3,2)	25/19	32/7
28	10	(8,0,4)	(16,0,8)/(19,0,2)	24/21	36/8
29	10	(9,1,5)	(19,1,6)/(21,0,1)	27/22	25/9
30	10	(8,2,3)	(20,2,3)/(20,0,2)	28/21	31/6
average	5	(3.1,1.2,1.9)	(14.5,2.5,8.5)/(16.4,0.5,3.0)	18.8/21.3	24.5/6.9

Table A.1: Results of benchmark testing

number of terms. The Hopfield net thresholds (θ_0 and θ_j), on the other hand, were robust enough to guarantee a reasonable number of hits.

Knowledge sources activated by the branch-and-bound algorithm appeared to be more strongly associated with the origins (knowledge sources) of the starting terms than those activated by the Hopfield net. For example, when using the branch-and-bound method, if the starting term was from the LCSH, then the final branch-and-bound suggested terms were more likely to be from the LCSH than from other sources. The Hopfield net, on the other hand, appeared to invoke the different knowledge sources more evenly. As shown in Table A.1, for most queries, the Hopfield net (NN) almost always produced terms from all three knowledge sources (i.e., more evenly), while the branch-and-bound (SN) often produced terms from only a couple of knowledge sources (usually identical to the sources of the query terms). We believe this was because the parallel relaxation process branched out to other knowledge sources more efficiently than the serial search, with the result that combining evidence from different activated nodes as implemented in Hopfield net activation caused more even activation of terms from all sources.

Appendix B

SAMPLE SESSIONS

Sample sessions of branch-and-bound (BAB) and Hopfield net (HP) spreading activations are presented below. Comments are enclosed in parentheses to indicate all the interactions between a subject and the system. The subject was requested to identify topics (with the help of the system) relevant to "KIDS: A Query and Inference System Based upon Knowledge Indexed Deductive Search," by K. Lee, a Georgia State University Ph.D. dissertation, 139 pages, 1989. An abstract of this dissertation was also presented to the subject. Details of the experiment are discussed in the the section on *User Evaluation* in Chapter 4.

```
*--------------------*
    Initial terms:  {* Supplied by the subject and used by both algorithms. *}
    -------------
 1. (P L) INFORMATION RETRIEVAL {* P: Public, A: ACM, L: LCSH *}
 2. (P  ) KNOWLEDGE BASE
 3. (P  ) THESAURUS
 4. (P L) AUTOMATIC INDEXING
*--------------------*
```

A. Branch-and-bound activation:

```
{* The subject selected the branch-and-bound search module first. *}

Enter the number of system-suggested terms or '0' to quit >> 30
{* User supplied his desired number of suggested terms. *}

{* The algorithm searched all three knowledge sources and suggested
```

```
   terms in decreasing order of relevance. Starting terms were included. *}
  1. ( ) THESAURUS
  2. ( ) INDEXING
  3. ( ) KEVIN.HOT               {* User-specific folder in Public DB. *}
  4. ( ) KNOWLEDGE BASE
  5. ( ) INFORMATION RETRIEVAL
  6. ( ) AUTOMATIC INDEXING
  7. ( ) DBMS.AI                 {* Terms with *.* are Public folder names. *}
  8. ( ) ROSS.HOT
  9. ( ) INFORMATION RETRIEVAL SYSTEMS
 10. ( ) RETRIEVAL
 11. ( ) EXPERT SYSTEMS
 12. ( ) INFORMATION
 13. ( ) DATABASE
 14. ( ) CARAT.DAT
 15. ( ) QUERY
 16. ( ) RECALL
 17. ( ) LANGUAGE
 18. ( ) SUPPORT
 19. ( ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 20. ( ) RESEARCH
 21. ( ) GQP.DAT
 22. ( ) MODEL
 23. ( ) KEYWORD
 24. ( ) PRECISION
 25. ( ) USER INTERFACES
 26. ( ) PETER.HOT
 27. ( ) ARTIFICIAL INTELLIGENCE
 28. ( ) MANAGEMENT
 29. ( ) EXPERT SYSTEM
 30. ( ) LOGIC
 31. ( ) OBJECT
 32. ( ) SEMANTIC.MDL
 33. ( ) DATABASE MANAGEMENT SYSTEMS
 34. ( ) EXPERT
 35. ( ) DESIGN

Enter numbers [1 to 35] or '0' to quit:  1, 2, 4-6, 9, 16, 19, 24
{* The subject selected desired terms. *}

{* The system listed the user-selected terms and their sources. *}
 1. (P  ) THESAURUS
 2. (P  ) INDEXING
 3. (P  ) KNOWLEDGE BASE
 4. (P L) INFORMATION RETRIEVAL
 5. (P L) AUTOMATIC INDEXING
 6. (P L) INFORMATION RETRIEVAL SYSTEMS
 7. (P  ) RECALL
 8. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 9. (P  ) PRECISION
```

```
Enter the number of system-suggested terms or '0' to quit >> 50
{* The subject used the selected terms to activate the branch-and-bound
   algorithm again. *}

{* More terms were suggested. *}
  1. ( ) KEVIN.HOT
  2. ( ) INDEXING
  3. ( ) INFORMATION RETRIEVAL
  4. ( ) RECALL
 ........
 54. ( ) DATA STRUCTURES
 55. ( ) PERFORMANCE
 56. ( ) QUERY.OPT
 57. ( ) ARTIFICIAL INTELLIGENCE
 58. ( ) KEYWORD
 59. ( ) THESAURI
 60. ( ) USER INTERFACES

{* More selections. *}
Enter numbers [1 to 60] or '0' to quit:  2-6, 8, 10, 11, 14, 15
{* Notice that terms were ranked in different order than had
   been suggested previously. This was because of the different
   starting terms. *}

 1. (P  ) INDEXING
 2. (P L) INFORMATION RETRIEVAL
 3. (P  ) RECALL
 4. (P  ) PRECISION
 5. (P L) INFORMATION RETRIEVAL SYSTEMS
 6. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 7. (P  ) THESAURUS
 8. (P L) AUTOMATIC INDEXING
 9. (  L) INFORMATION STORAGE AND RETRIEVAL SYSTEMS {* A new term. *}
10. (P  ) KNOWLEDGE BASE

Enter the number of system-suggested terms or '0' to quit >> 0
{* The subject decided to stop the search process. *}

{* A total of 10 terms were selected. Six terms were suggested by the
   branch-and-bound algorithm and they came from two knowledge sources:
   the Public KB and the LCSH. *}
```

B. Hopfield net activation:

```
{* The subject selected the Hopfield net search module.
```

```
   Similar interface was used for the Hopfield net search process. *}

Enter the number of system-suggested terms or ‘0’ to quit >> 10
{* Notice that the suggested terms were different from those suggested
   by the branch-and-bound algorithm. *}

  1. ( ) INDEXING
  2. ( ) SELLING - INFORMATION STORAGE AND RETRIEVAL SYSTEMS
  3. ( ) KEVIN.HOT
  4. ( ) INFORMATION RETRIEVAL SYSTEM EVALUATION
  5. ( ) RECALL
  6. ( ) EXPERT SYSTEMS
  7. ( ) THESAURUS
  8. ( ) DBMS.AI
  9. ( ) ROSS.HOT
 10. ( ) INFORMATION STORAGE AND RETRIEVAL SYSTEMS
 11. ( ) INFORMATION RETRIEVAL
 12. ( ) KNOWLEDGE BASE
 13. ( ) AUTOMATIC INDEXING

Enter numbers [1 to 13] or ‘0’ to quit:  1, 2, 4, 5, 7, 10-13

 1. (P  ) INDEXING
 2. (  L) SELLING - INFORMATION STORAGE AND RETRIEVAL SYSTEMS
 3. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 4. (P  ) RECALL
 5. (P  ) THESAURUS
 6. (  L) INFORMATION STORAGE AND RETRIEVAL SYSTEMS
 7. (P L) INFORMATION RETRIEVAL
 8. (P  ) KNOWLEDGE BASE
 9. (P L) AUTOMATIC INDEXING

Enter the number of system-suggested terms or ‘0’ to quit >> 30
 ........

Enter number [1 to 40] or ‘0’ to quit:  3-7, 9, 33, 35, 36, 38
 ........

Enter numbers [1 to 67] or ‘0’ to quit: 0
{* The system listed his final selections. *}

 1. (P  ) PRECISION
 2. (P L) INFORMATION RETRIEVAL
 3. (P  ) INDEXING
 4. (P L) AUTOMATIC INDEXING
 5. (P  ) RECALL
 6. (  L) AUTOMATIC ABSTRACTING                {* Suggested by HP, not BAB. *}
 7. (  L) AUTOMATIC CLASSIFICATION             {* Suggested by HP, not BAB. *}
 8. (  L) AUTOMATIC INFORMATION RETRIEVAL      {* Suggested by HP, not BAB. *}
 9. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
```

```
10. (P  ) THESAURUS
11. (  L) INFORMATION STORAGE AND RETRIEVAL SYSTEMS
12. (P  ) KNOWLEDGE BASE

{* A total of 12 terms were selected. Eight terms were suggested by the
   Hopfield net algorithm. Terms 6, 7, and 8 were different from those
   suggested by the branch-and-bound algorithm. *}
```

Appendix C

CANCER SPACE: A WEB-BASED INFORMATION RETRIEVAL SYSTEM

Section 5.6.3 describes how a large-scale concept space was generated from approximately one million CancerLit document records. In order to conduct experiments related to research in Digital Library and Medical Informatic as well as provide public access to both CancerLit library and its concept space, a web-based information retrieval system, *Cancer Space*, was built. *Cancer Space* can be found at `http://ai20.bpa.arizona.edu/cgi-bin/cancerlit/cn/`

Although traditional information retrieval systems and web-based search engines have been commercially or freely available, their keyword-based retrieval architectures do not support the needs for indexing and searching concept spaces. A new concept space search engine was designed based on a client-server architecture, which relies on a Common Gateway Interface (CGI) access through a Hypertext Transfer Protocol (HTTP). A web-CGI architecture provides fast system development because of the readily usable network protocol from web servers and graphical user interface from web browsers. During four years of development and use of concept space search engine, various servers have been built on a

wide range of platforms ranging from laptops to supercomputers in both Unix and Windows operating systems.

C.1 **Client User Interface**

The client-side implementation provides an interface for users to input their queries and for the corresponding search engine to display search results. The basic design of the user input interface is built on *FORM* elements of Hypertext Markup Language (HTML) (http://www.w3.org/). Initially, users can type in their query terms (one or more), select a *search space*, and submit their queries by clicking on a button.

There are currently two *search spaces*: *Concepts* and *Documents*. Concept search returns *co-occurred terms* from concept space for contextual query expansion, while document search retrieves the actual documents that match the user's query.

- *Concept Space:*

  In concept space search, the interface displays a list of co-occurred terms ranked in a non-decreasing order according to their co-occurred weight in up-to-three columns depending on the types of concept. Currently, the CancerLit concept space carries three types: *Noun Phrase* for terms extracted

from free-text, *MeSH* for Medical Subject Headings assigned to each document, and *Author* of documents. The most relevant terms (or concepts) are listed first in their type categories. Users can use the returning list of concepts to help them to further refine or change their search.

Users can select concepts from the co-occurred list and to augment or alter their queries. Users can submit their refined queries to the concept space search and find out what other concepts are closely related to concepts in their query sets. Users can iterate through such a process to find a set of concepts, which best describes their target concept. Such iterative process serves as query refinement to a set of search terms.

Figure C.1 shows a concept space search results from two terms, "Cancer Surgery" and "routine care". For the purpose of backtracking, search terms, which activate concepts appearing in the table, are enumerated by alphabets. For visually emphasizing concepts coming from multiple search terms, color bars are drawn after the first group of having the most number of search terms.

- *Document Space:*

  In document space search, the interface shows a list of relevant documents grouped by the number of matched terms and ranked in a non-decreasing order according to weight calculated by the vector space model. Users have

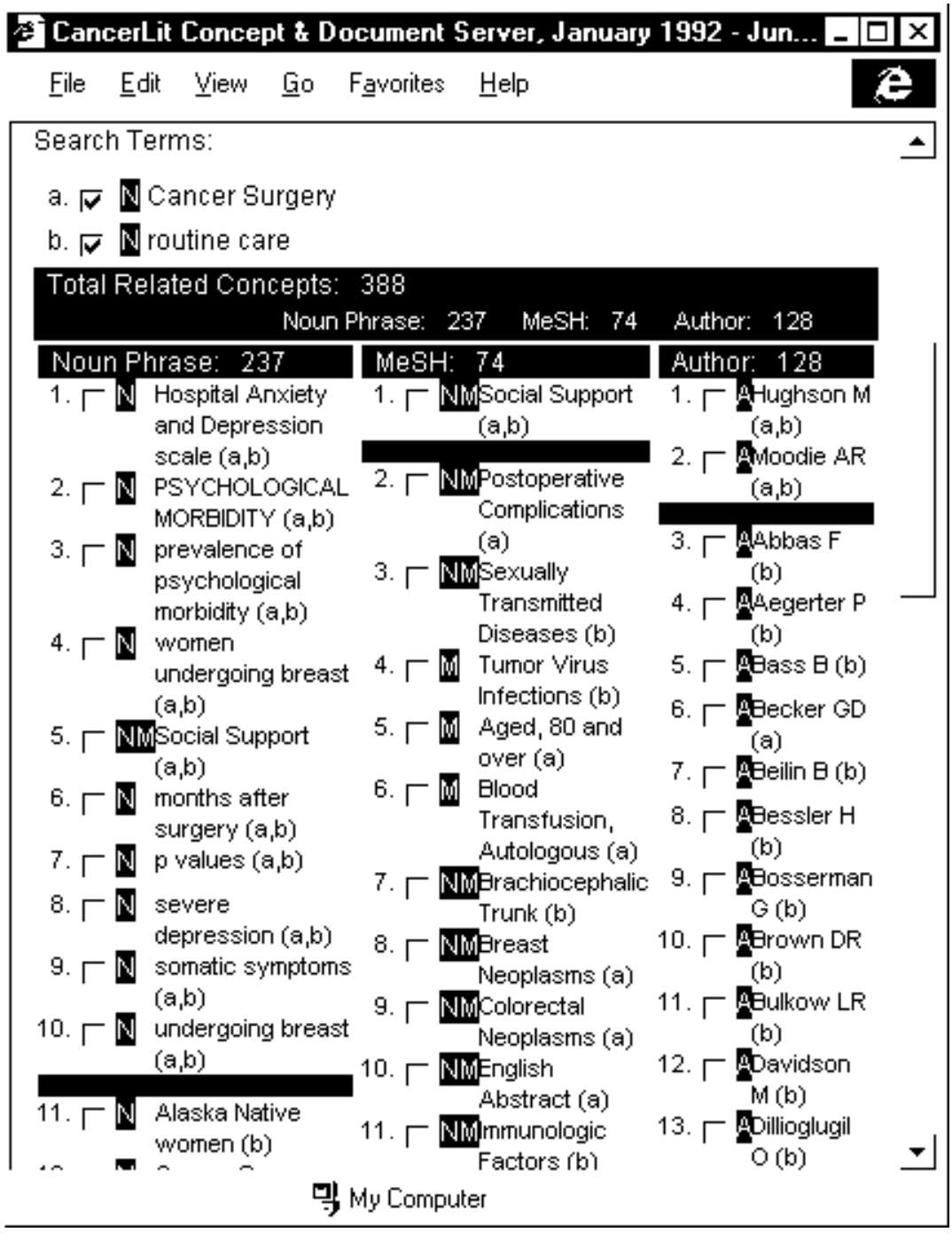

Figure C.1: Cancer Space: Results from concept space search

a choice of requesting *Citations* or *Abstracts with Citations* to view document records. The *Citation* shows only *title*, *author*, and *source* of each retrieved document. The *Abstracts with Citations* shows extra information like *MeSH term* assignments and *abstract* to each document.

Figure C.2 shows the document retrieval results from two search terms: "Cancer Surgery" and "routine care". Near the top right corner, it displays document counts for matching both and just one of the two search terms. In this retrieval, two documents have both search terms and ninety-five have just one.

C.2 Server Implementation

The server-side implementation provides a mechanism to store and retrieve information for query input. The storage component is built as a read-only database system, which had several indexing scheme to speed up the retrieval process for all requests. The server uses traditional *keyword* inverted index to index both automatic thesaurus and documents. The difference is that *term phrases* or *concepts* were used instead of single-word keywords as "inverted units" to index their corresponding co-occurred terms and documents.

The retrieval component implemented is similar to most other search engines on the web. The CGI environment provides input (from user query) to a search engine through the HTTP server sitting on our local machine. For both concept

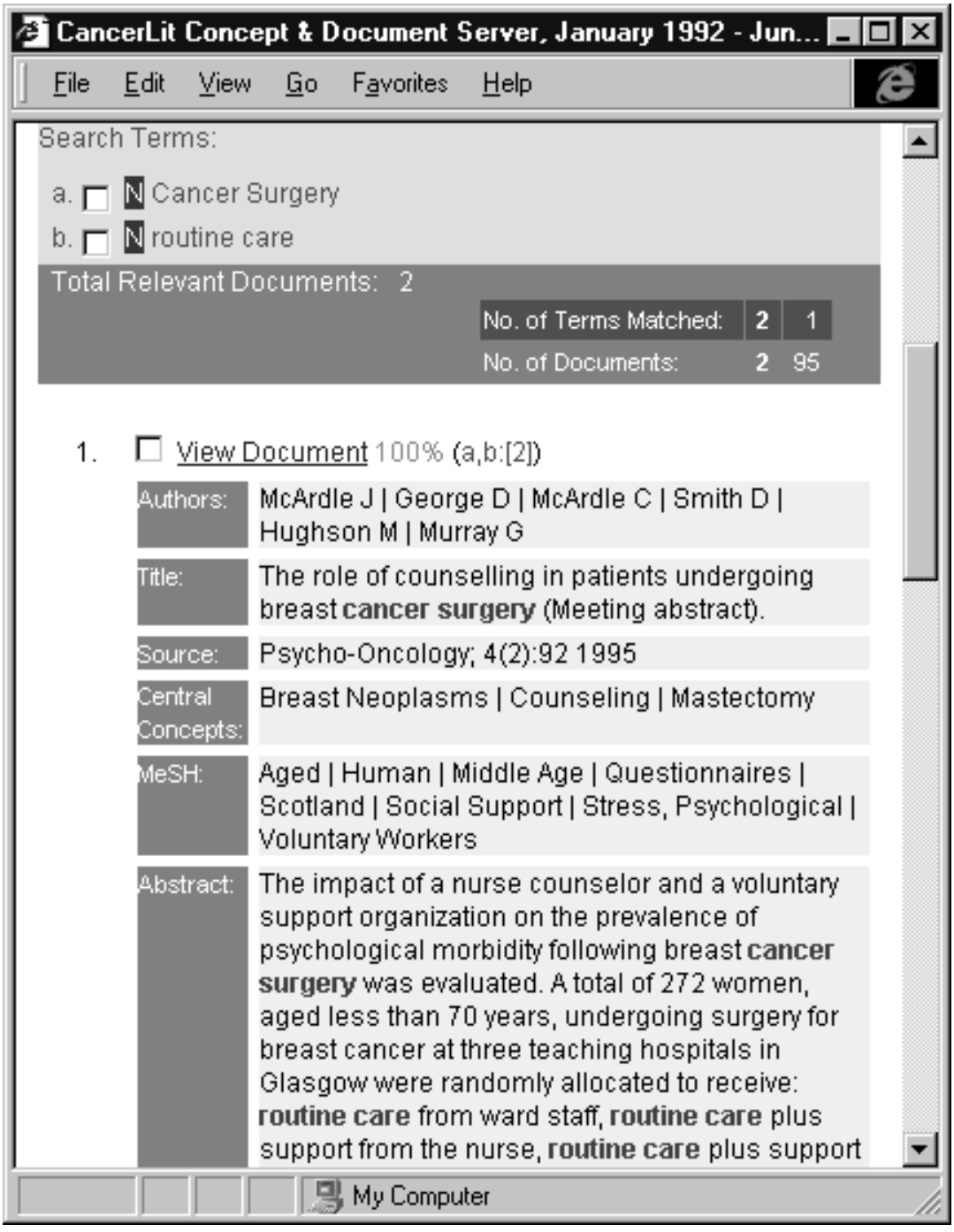

Figure C.2: Cancer Space: Results from document retrieval

and document searches, all input terms are used to match against a list of searchable terms in the server. The matching process does some simple stemming and normalization of terms in the same manner in which the thesaurus was generated in order to better connect queries with documents. If at least one input term matches the searchable list, a search of either the concept or document space will be conducted. If no terms match the user's query, the server will issue a message notifying the user of an unsuccessful search. The server formats both results for concept and document spaces in HTML format as dynamic web pages and delivers it to the users' browsers.

Appendix D

FUNDING AND ACKNOWLEDGMENTS

D.1 Concept Space Consultation

This project was supported mainly by a National Institutes of Health (NIH) BRSG grant #S07RR07002 and a National Science Foundation (NSF) grant #IRI-9211418. We would also like to thank K. Basu, K. Ng, C. Wei, J. Martinez, K. Leung, and R. Orwig for their involvement in system development and evaluation.

D.2 Concept Space Generation

This project was supported mainly by the following grants: (1) NSF/ARPA/NASA Digital Library Initiative, IRI-9411318, 1994-1998 (B. Schatz, H. Chen, et al., "Building the Interspace: Digital Library Infrastructure for a University Engineering Community"), (2) NSF CISE Research Initiation Award, IRI-9211418, 1992-1994 (H. Chen, "Building a Concept Space for an Electronic Community System"), (3) NSF CISE Special Initiative on Coordination Theory and Collaboration Technology, IRI-9015407, 1990-1993 (B. Schatz, "Building a National Collaboratory

Testbed"), (4) AT&T Foundation Special Purpose Grants in Science and Engineering, 1994-1995 (H. Chen), and (5) National Center for Supercomputing Applications (NCSA), High-performance Computing Resources Grants, 1994-1996 (H. Chen).

We would also like to thank Dr. Anindya Datta, Pauline Cochrane, and Dr. Ann Bishop for their valuable suggestions, Jim Ashling of the Institution of Electrical Engineers (IEE, vendor of the INSPEC database) for providing the INSPEC thesaurus for our experiment, and Dr. Larry Smarr, Dr. Melanie Loots, and Mike Welge at NCSA for their kind assistance.