To the Graduate Council:

I am submitting herewith a thesis written by David K. Trubey entitled "JANE: A New Information Retrieval System for the Radiation Shielding Information Center." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfullment of the requirements for the degree of Master of Science, with a major in Computer Science.

Maria Zemankova

Maria Zemankova, Major Professor

We have read this thesis

and recommend its acceptance:

Accepted for the Council:

Vice Provost

and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under the rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in her absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature David K. Imbey

Date May 27, 1988

JANE: A NEW INFORMATION RETRIEVAL SYSTEM

FOR THE

RADIATION SHIELDING INFORMATION CENTER

A Thesis Presented

for the

Master of Science Degree

The University of Tennessee, Knoxville

David K. Trubey

August 1988

To Jean Marie

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Maria Zemankova, my major professor, for her assistance and recommendations throughout this project and especially for her many valuable suggestions for the thesis itself. I am also grateful to the other members of my Master's Committee, Dr. Bruce W. Char and Dr. David W. Straight, for their valuable suggestions. I would also like to thank Dr. Robert W. Roussin, Director of the Oak Ridge National Laboratory Engineering Physics Information Centers, for his encouragement and support.

The system developed for this project is named JANE in honor of Jane Gurney who conscientiously managed the RSIC information system from the time it became operational on the Oak Ridge IBM 7090 machine in the early 1960s until it was in routine use on the IBM 3033 computer. During the early period, the system included a Selective Dissemination of Information (SDI), which involved many complications. This included managing a frequently changing file of user interest profiles.

I would also like to acknowledge the excellent work of Ann Gustin who has managed the RSIC literature flow for many years. This includes ordering the literature, keeping records, and entering citation information which is published in the *RSIC Newsletter* and placed in the database. It also includes the chore of persuading procrastinating staff members to review abstract journals and other documents.

This thesis would not have been possible without sacrifice on the home front. My deep appreciation goes to my wife, Jean, for her encouragement over the long period that I worked on my degree, especially while so many weekends were diverted to this project.

ABSTRACT

A new information storage and retrieval system has been developed for the Radiation Shielding Information Center (RSIC) at Oak Ridge National Laboratory to replace mainframe systems that have become obsolete. The database contains citations and abstracts of literature which were selected by RSIC analysts and indexed with terms from a controlled vocabulary. The database, begun in 1963, has been maintained continuously since that time.

The new system, called JANE, incorporates automatic indexing techniques and on-line retrieval using the RSIC Data General Eclipse MV/4000 minicomputer. Automatic indexing and retrieval techniques based on fuzzy-set theory allow the presentation of results in order of Retrieval Status Value. The fuzzy-set membership function depends on term frequency in the titles and abstracts and on Term Discrimination Values which indicate the resolving power of the individual terms. These values are determined by the Cover Coefficient method. The use of a commercial database base to store and retrieve the indexing information permits rapid retrieval of the stored documents.

Comparisons of the new and presently-used systems for actual searches of the literature indicate that it is practical to replace the mainframe systems with a minicomputer system similar to the present version of JANE.

TABLE OF CONTENTS

SECTION	PAGE
1. INTRODUCTION	1
1.1 Background	1
1.2 History of RSIC Database	2
1.3 Present Status of Database	3
1.4 Purpose and Approach	7
2. REVIEW OF LITERATURE	10
2.1 Information Retrieval Systems	10
2.1.1 General Reviews	10
2.1.2 Query Interpretation	12
2.1.3 Vector Space Model	13
2.1.4 Application of Database System Techniques	14
2.2 Automatic Indexing and Retrieval	14
2.3 Fuzzy-Set Applications to Information Retrieval	17
2.4 Artificial Intelligence	19
3. ALGORITHMS ADOPTED FOR JANE SYSTEM	22
3.1 Algorithms Adopted for Automatic Indexing	22
3.1.1 Document-Term Assignment	22
3.1.2 Document-Term Frequency Assignment	24
3.1.3 Term Discrimination Value	25
3.2 Algorithms Adopted in Document Retrieval	29
4. DEVELOPMENT OF THE NEW SYSTEM	34
4.1 Requirements	34
4.2 Resources	35

SECTION PAGE

4.3 Implementation Overview 35

4.3.1 System Organization 35

4.3.2 Data Structures 36

4.3.3 Conversion to the Eclipse 38

4.3.4 Development of the INFOS II Files 38

4.3.5 The JANE Retrieval Program 39

5. JANE SYSTEM 42

5.1 Overview 42

5.2 User-Program Dialog 43

5.3 Limitations 49

6. COMPARISON OF NEW AND PRESENT SYSTEMS 53

6.1 ADSEP System 53

6.2 ORLOOK Program 54

6.3 Results of Searches 55

6.4 Conclusions from Systems Comparison 65

7. CONCLUSIONS 68

7.1 Summary 68

7.2 Further Development 69

LIST OF REFERENCES 71

APPENDICES 77

APPENDIX A 78

APPENDIX B 83

VITA 85

LIST OF FIGURES

FIGURE		PAGE
1.	ADSEP-ORLOOK System Organization.	5
2.	Sample Set of RSI Subject Categories.	6
3.	Sample RSI Document in ADSEP Format.	8
4.	JANE System Organization.	37
5.	Main Menu with Selection of Option 1.	44
6.	Use of Option 2.	46
7.	'Combine Sets' Option.	48
8.	Print Records Option.	50
9.	Print Menu and Output	51
10.	JANE Retrieval Program Subroutine Structure.	79

LIST OF ACRONYMS

ADSEP	IBM mainframe database maintenance program	3
AI	Artificial intelligence	19
CC	Cover Coefficient	15
DG	Data General	35
DOE	Department of Energy	1
IR	Information Retrieval	22
INFOS II	Data General database system	35
ORLOOK	IBM mainframe information retrieval program	1
ORNL	Oak Ridge National Laboratory	1
RECON	IBM mainframe information retrieval program	1
RSC	Radiation Shielding Codes database	7
RSI	Radiation Shielding Information database	7
RSIC	Radiation Shielding Information Center	1
SDI	Selective Dissemination of Information	2
RSV	Retrieval Status Value	19
SORT/MERGE	Data General sort and merge utility	35
TDV	Term Discrimination Value	10
TSO	Time Sharing Option, IBM time-sharing system	3

1. INTRODUCTION

1.1 Background

The Radiation Shielding Information Center (RSIC) at Oak Ridge National Laboratory (ORNL) collects, organizes, analyzes, and distributes information on radiation shielding, radiation transport, and radiation protection [Tr67]. The major portion of its work is the collection, testing, packaging, and distribution of computer codes and computer-readable data libraries used for radiation analysis calculations. The codes are tested on computers ranging in size from desk-top personal computers through minicomputers and mainframes to supercomputers and made available world-wide.

At the time of its founding, it was realized that a computerized database to give access to the literature would be useful in maintaining awareness of past research. Such a database was started and has been an ongoing activity of the Center ever since. In recent years, the database has been maintained on the ORNL IBM mainframes and was searchable from remote terminals nation-wide using the U.S. Department of Energy RECON system [DOE83]. On January 1, 1987, the RECON system at ORNL was shut down, and the RSIC database became unavailable to outside users. RECON was shut down because the Department of Energy decided that the system was outmoded and too expensive to maintain. The main function of RECON, from the DOE perspective, was to make the DOE Energy database available. This function has been transferred to commercial vendors.

Since it was necessary for RSIC staff to access its database routinely to respond to information requests, it was decided to use a readily-available search program which had been developed in the early 1970s. This program, ORLOOK [Si75] can be used by ORNL staff to search databases which employ the same data structures as did RECON. The use of this system is less than satisfactory for several reasons, and therefore, an alternative system is now being sought.

1.2 History of RSIC Database

The RSIC data base, consisting of citations and abstracts of literature on radiation shielding and transport collected by the ORNL Radiation Shielding Information Center, was begun in 1963. The original program, probably the first information retrieval program implemented at Oak Ridge [Pe63], was written for the IBM 7090, a machine with a 32K, 36-bit-word memory and many tape drives. The programming language was Fortran II, but the program required some assembly language routines. The data input device was a punched-card reader. Inverted files were stored on magnetic tape which had to be rewritten with each update. Searches involved much tape "spinning." At that time, most information organizations at ORNL and elsewhere were using manual or mechanical methods for information retrieval [Ke62].

The controlled, precoordinated indexing language developed in 1963 has remained essentially the same, although it has been refined over time. A 3-level weight (emphasis) was associated with each indexing term (of course, fuzzy sets were unknown to the developers). The weight was intended to represent the degree to which the term was emphasized in the document. Additional, unusual features were indicators as to whether a document provided an exposition of a topic or provided comparisons of results to other work.

The first major revision to the program incorporated a procedure for a selective dissemination of information (SDI) to people on the RSIC distribution list. With each update, a list of abstracts, corresponding to a person's interest profile, was mailed. These profiles were constructed from the response to a questionaire provided to people requesting the SDI service. If this service were to be revived, an interesting research project might be to develop an automatic profile construction procedure using artificial intelligence techniques.

In addition, general bibliographies were published from time to time, and searches were performed upon request.

The system was completely reprogrammed in Cobol in 1969 [Jo69, Jo71] when the IBM System 360 became available. The new use of disk drives provided a marked increase in efficiency, although the system was still entirely operated in a batch mode. Use of the IBM magnetic tape typewriter

increased input efficiency, and it allowed the use of upper and lower case, since the IBM 360 had an 8-bit byte for the first time. The typewriter cartridge had to be processed by a special purpose machine (Digi-Data Converter) to change code to EBCDIC and write to a standard computer tape.

The next major change occurred in 1979, when it was decided to convert the data to a different format so that it could be incorporated as a database in the Department of Energy interactive RECON system [DOE83]. Both DOE RECON and NASA RECON were developed by Lockheed as precursors to Lockheed's well-known commercial Dialog system. The RECON system, reprogrammed at ORNL because Lockheed did not provide the source code, was accessible to DOE contractors and others nationwide. By the time RSIC converted to the use of RECON, it was available to users in a dial-up mode using ASCII terminals.

1.3 Present Status of Database

On January 1, 1987 DOE RECON was shut down for budgetary reasons as mentioned above; RECON had become too costly to maintain. This left RSIC with no means to search its files, but the situation was resolved for the short term by using the ORLOOK program running under TSO, one of the time-sharing control programs on the IBM mainframe. ORLOOK, developed about 1970 [Si75], can search bibliographic files, using string matching, and perform the usual set operations needed for information retrieval. This solution is not satisfactory because ORLOOK is not cost effective; it is slow, output formats are fixed and not desirable, and it is limited to searching a 10 000-document file. The RSIC database is now stored in four files, and it is cumbersome to select a new file, which must be done several times to search the entire database. No inverted files are maintained, hence efficient retrieval based on indexing cannot be used. There is no possibility of upgrading ORLOOK in any way.

The RSIC database is now maintained by use of ADSEP (Automatic Data Set Entry Program) [Br70] in a batch mode. ADSEP is a database maintenance program which was developed at Oak Ridge about 1970 for the IBM mainframe computers. A number of retrieval programs can access ADSEP databases, including RECON. Although RECON is not now available, ADSEP is still being used, but the files are accessed by other programs. In the years since 1970, commercial database systems have been installed on Oak Ridge mainframe computers, but none proved as

useful for information retrieval purposes as ADSEP and the many programs which accessed ADSEP databases. Fig. 1 shows the overall organization of the ADSEP-ORLOOK system.

The main citation data, such as titles, authors, and journal references, are created on the RSIC Data General Eclipse MV/4000 computer by an RSIC staff member as new documents become known. Documents not already possessed by RSIC are placed on order when their existence becomes known. Information about current literature comes from many sources. The RSIC staff reviews technical journals, abstract journals, published bibliographies, technical reports, dissertations, conference proceedings, publishers' announcements, and receives reprints from authors. These citations are published in the monthly RSIC Newsletter by automatically inserting typesetting commands and running a program which generates input data for a photo-typesetter.

The data are also entered into the database on the ORNL IBM 3033 mainframe system in the following way. The RSIC computer is connected by network to the IBM mainframe system, and this allows the RSIC staff member to submit a "job" consisting of control information and input data to run the ADSEP program and update the database.

As documents are received, either from institutions which have placed RSIC on a mailing list, authors who send unsolicited reprints, or from previously placed orders, they are given to nuclear engineers for review. As time permits (the nuclear engineers who review the documents have many other priorities), the literature is reviewed and subject category indexing codes are assigned to the documents that are judged suitable for inclusion in the collection. The indexing codes represent a form of precoordinated indexing, that is, they represent the intersection of a set of simple keywords. As such, they represent nuclear engineering concepts. This is intended to increase retrieval efficiency and make the indexing data for a document more compact. A sample display of the subject category codes and their logically-equivalent keywords are shown in Fig. 2.

The indexing information and abstracts are then keyed into a central DecSystem-10 computer by a data entry service group. The files produced by the service group are processed to check for validity of the indexing codes, and keywords are incorporated automatically, based on the indexing codes. The output file from this processing is then entered into the database by running an ADSEP job.

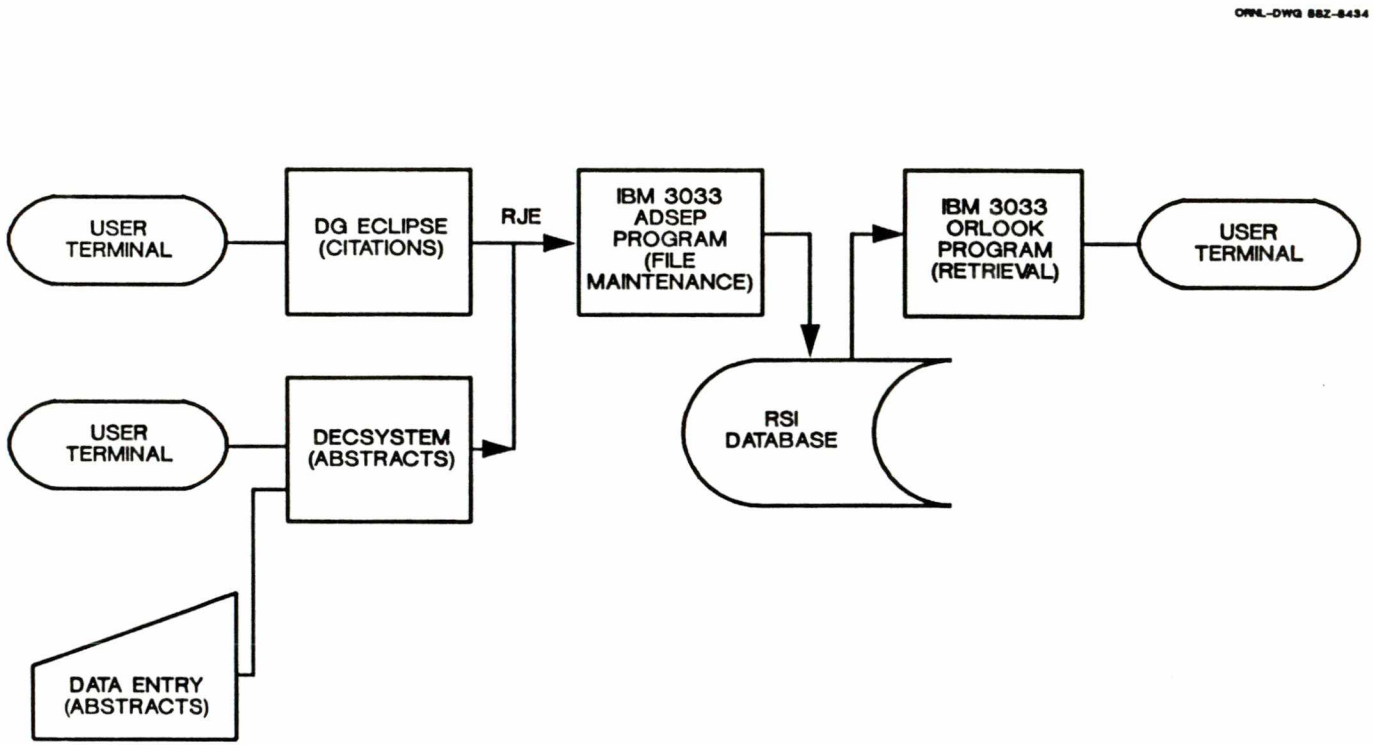


Figure 1. ADSEP-ORLOOK System Organization

GENERAL RADIATION SHIELDING REFERENCES

000000 General Radiation Shielding References
010000 Review Articles
020000 Standards and Benchmark Problems

BASIC NUCLEAR AND ATOMIC INFORMATION

111110 Sources - Neutron - Fission or Reactor - Prompt
111120 Sources - Neutron - Fission - Delayed
111130 Sources - Neutron - Fission - Weapon
111210 Sources - Neutron - Reaction - 14 MeV & D,T Fusion
111220 Sources - Neutron - Reaction - Spectral
111300 Sources - Neutron - Activation
111400 Sources - Neutron - Thermonuclear Weapon
111500 Sources - Neutron - Discrete Energies (Calculation)
111600 Sources - Neutron - Reaction - Photoneutron (in shield)
111700 Sources - Neutron - Mixed Oxide Fuels and Spent Fuel
112A00 Sources - Gamma-Ray - Secondary Radiation from Weapons
112B00 Sources - Gamma-Ray - Discrete Energies (Calculation)
112110 Sources - Gamma-Ray - Fission - Prompt
112120 Sources - Gamma-Ray - Fission-Product-Decay (including Yields)
112130 Sources - Gamma-Ray - Fission - Weapon (Initial)
112200 Sources - Gamma-Ray - Bremsstrahlung
112300 Sources - Gamma-Ray - Annihilation
112400 Sources - Gamma-Ray - Neutron-Inelastic-Scattering
112500 Sources - Gamma-Ray - Neutron-Capture
112600 Sources - Gamma-Ray - Neutron-Reaction-Product (n,xγ) Charged Particles
112710 Sources - Gamma-Ray - Radionuclide-Decay - General Surveys
112720 Sources - Gamma-Ray - Radionuclide-Decay - ^{60}Co
112730 Sources - Gamma-Ray - Radionuclide-Decay - ^{137}Cs
112740 Sources - Gamma-Ray - Radionuclide-Decay - ^{198}Au
112750 Sources - Gamma-Ray - Radionuclide-Decay - ^{24}Na
112760 Sources - Gamma-Ray - Radionuclide-Decay - Other (includes corrosion products)
112770 Sources - Gamma-Ray - Radionuclide-Decay - ^{16}N
112800 Sources - Gamma-Ray - Thermonuclear Weapons
112910 Sources - Gamma-Ray - Fallout from Weapons
112920 Sources - Gamma-Ray - Simulated Fallout from Weapons
112930 Sources - Gamma-Ray - Reactor Accident (airborne, fallout, or plume sources)
112940 Sources - Gamma-Ray - Mixed Oxide Fuels and Spent Fuel

Figure 2. Sample Set of RSI Subject Categories

There are now more than 16 000 records (document citations) in the database which was known on RECON as RSI. The block of data called an ADSEP record is a surrogate for a published document. It can be accessed by number, frequently called the ADSEP number or accession number, and contains labeled fields such as title, authors, abstract, and keywords. Hereafter, we will refer to such a data block as a "document." A sample document in ADSEP format is shown in Fig. 3.

Another RSIC database, called RSC, consisting of references to literature on computer codes, was also available on RECON, and a new system developed for the RSI database could easily be adapted for use with RSC.

In addition to cataloging fields, such as title, authors, and date of publication, the RSI records have two fields which contain indexing information: subject category codes (mentioned above) and keywords which are subsequently generated automatically from the subject categories. These are added to the ADSEP record (document) to make searching easier for users who are not familiar with the precoordinated subject category codes.

1.4 Purpose and Approach

The purpose of this research project was to develop an information retrieval system that can use the database as it now exists and replace the present outmoded IBM mainframe retrieval system. The new system uses automatic indexing techniques to replace the costly manual indexing now being used and employs fuzzy-set concepts in connection with both indexing and retrieval. The project demonstrates that it is practical to move the RSIC information storage and retrieval function to the RSIC minicomputer.

The first step in developing the system was to review the the current literature, emphasizing papers on automatic indexing, fuzzy-set theory applications to information retrieval, and artificial intelligence applications. The literature that was of special interest is reviewed in Sect. 2.

Several algorithms found in the literature were adapted for this project and several others were devised by the author in the course of the system's development. These are explained in Sect. 3.

<110203>
<ABSTRACT>In order to assess group constants for reactor materials, energy spectra of neutrons from about 1 keV to a few MeV in some sample assemblies have been measured by the linac time-of-flight method and the results are compared with those theoretically predicted by one-dimensional transport calculation. In addition to the ordinary large bulk pile system, a smaller pile surrounded by a reflector and a slab scattering method have been investigated by introducing the sensitivity coefficients of group constants to neutron spectrum. Typical results for some reactor materials taken at the Research Reactor Institute, Kyoto University are shown.
<AUTHOR >Kimura, I.
<AVAIL >Supt. of Documents, U.S. Government Printing Office, Washington, D.C. 20402
<BIB >1
<CAT EMP >111110 1; 122110 3; 32A000 1; 521230 3; 810000 3; 795254 3
<CAT NUM >111110; 122110; 32A000; 521230; 810000; 795254
<CORPAUTH>JAPAN
<DOC TYPE>C
<EDITION >8
<KEYWORDS>SOURCE; NEUTRON; FISSION; REACTOR SOURCE; PROMPT; INTERACTIONS OF RADIATION WITH MATTER; GROUP CROSS-SECTIONS; CARLSON SN; DISCRETE-ORDINATES; INTERFACE EFFECTS; LAMINAR LAYERS; EXPERIMENT; MATERIALS (GENERAL SURVEY)
<PUB DATE>September 1980
<PUB DESC>NBS Spec. Pub. 594, pp. 265-274; CONF-791058, pp. 265-274
<SRCHDATE>800900
<SUB CAT >111110 01; 122110 03; 32A000 01; 521230 33; 810000 03; 795254 03
<TITLE >Measurement and Analysis of Neutron Spectra in Some Assemblies of Reactor Materials.

Figure 3. Sample RSI Document in ADSEP Format

The implementation of the newly-developed JANE system is described in Sect. 4. Programs were developed to perform the automatic indexing of the author names and terms in the titles and abstracts of a representative portion of the RSIC database. The Data General INFOS II database system was used to store and retrieve the indexing information. Details of the JANE retrieval program and sample retrieval sessions are given in Sect. 5

The system was tested with a sample set of queries received by RSIC. These tests and the results are given in Sect. 6. Finally, the overall conclusions, based on testing the JANE system, are given in Sect. 7.

2. REVIEW OF LITERATURE

In order to formulate a plan for development of a new system, the literature was consulted to obtain ideas which might prove useful. The articles of most utility addressed the following themes: the present state of information retrieval technology, automatic indexing, fuzzy set logic, incorporation of artificial intelligence, and improved presentation of search results.

2.1 Information Retrieval Systems

2.1.1 General Reviews

Any review of results in information retrieval will reference the texbook by Salton and McGill [Sa83]. This book reviews information storage and retrieval as it existed in 1981 from the perspective of one of the leading research figures in the field. No more recent monograph is known to be available.

In their book, Salton and McGill discuss the concept of Term Discrimination Value (TDV) which was introduced by Salton in 1975 [Sa75]. This quantity can be computed for an indexing term as a measure of its ability to distinguish documents from each other, i.e., it is a measure of a term's "resolving power" with respect to a document, and hence is useful to determine its value as a term in the indexing language. The use of TDVs is discussed in more detail in Sect. 3.1.3 in connection with the implementation of the new system.

In regard to automatic indexing evaluation, one of the goals of this thesis research, Salton and McGill point out that comparative tests of indexing effectiveness (and efficiency) must normally be carried out under controlled conditions, using test collections of relatively small size with small sets of test queries. That is, many tests require detailed knowledge of the collection which is not possible for large, operational collections. For example, to compute the value of recall, the fraction of relevant documents in a collection retrieved by a particular query, one must know how many documents in the total collection are relevant to the query. This is impossible to know in a large collection. Furthermore, large collections are frequently not available for research use and so small, contrived databases are used. Although retrieval effectiveness of many automatic systems has been shown to be as good as manually operated systems, the authors state that it is hazardous to extrapolate test results to operational situations such as the Radiation Shielding Information Center.

A short review article, "Information Retrieval," by Brian Vickery of the University College, London, has recently been published in the Encyclopedia of Physical Science and Technology [Vi87]. Vickery briefly discusses most of the features available in present systems and the issues related to their design.

An excellent overview of information retrieval issues has been published by Bärtschi [Ba85]. He lists research areas as (1) Content analysis, (2) Information structures, and (3) Query evalution. The article concentrates on (2) and (3). The discussion of query evaluation covers the issues related to the three most important models in use today: weighted Boolean, vector space, and probabilistic retrieval. The model implementded in the JANE system, described in Sect. 3, can be considered to be weighted Boolean.

In a recent Annual Review of Information Science and Technology, Boyce and Kraft [Boy85] write on "Principles and Theories in Information Science." They point out that information retrieval models are attempts to relate representations of questions or queries for information to representations of records in a database, usually in terms of bibliographic references to documents. The main difference between database systems in general and retrieval systems is in the inherent fuzziness of this relationship, which is generally called "relevance." The matter is complicated in that there are several views of what it means for a document to be relevant to a query.

The sections in Boyce and Kraft most useful for our purposes are titled "Representations for Retrieval" and "Models of Information Retrieval." They state that the characteristics of an indexing language have little effect on performance and that simple natural language descriptions are just as useful as any complex artificial structure. This would imply that automatic indexing, with a reliance on the terms found in the title and abstract, would be as effective as the precoordinate indexing language used by RSIC for many years. This prospect provides considerable incentive for the present study.

2.1.2 Query Interpretation

Croft [Cr86] presents an approach to integrating Boolean and statistical systems where Boolean queries are interpreted as a means of specifying term dependencies in the relevant set of documents. He found that incorporating term dependencies derived from queries in a probabilistic retrieval model can improve retrieval performance.

In a recent paper, Das-Gupta [Da87] points out the ambiguous role of the conjunction "and" in natural language and the problems of correct Boolean interpretation. For example, the phrase "cats and dogs" introduces two instances of the more global concept "animals," whereas the phrase "antibiotics and sleeping sickness" introduces a single concept, the cure of sleeping sickness using antibiotics. The study resulted in an algorithm which utilizes semantic information and some syntactic information to obtain the appropriate Boolean interpretation. This problem with logical AND should be kept in mind by the user of most systems, including JANE.

The problem of negation in queries has recently been discussed by Eastman and Nakkouzi [Ea87]. They point out that the number of documents indexed under a specific term may be large, but the number which are *not* indexed under that term is likely to be unmanageable. Therefore, it is generally recommended that the NOT operator be used only to narrow down a set of items already obtained rather than as part of the original specification. In other words, it is used in the form of AND NOT. This recommendation can be regarded as an informal method of providing a local context in which to view the negation. They mention that psychological and linguistic studies indicate that people generally use negation in a local context rather than in a global context. For the above reasons, NOT is interpreted in the JANE system as being the set difference operator.

An innovative way of augmenting a retrieval system is to develop user profiles, such as those used for selective dissemination of information (SDI) systems, and use them to assist user query construction. Queries are frequently poorly formulated leading to poor retrieval results. Making use of user information, however, might lead to the automatic elimination of large sections of the database not of interest to the user, and to augmentation of the query by the inclusion of terms drawn from the profile. Shepherd and Phillips [Sh86] developed such a system and examined the relationship of clusters of documents that are retrieved by the profile and by the query and found that cluster similarity and cluster overlap can be expressed as linear functions of the similarity

existing between the original pairs of profiles and queries. This idea might be useful as a future upgrade to the JANE system.

2.1.3 Vector Space Model

It is common in information retrieval (IR) to represent each document by means of *keywords* or *index terms*. These are usually derived from the text or some surrogate (e.g., abstract) through a process of *indexing*. In addition to the selection of terms to represent the documents, it is common to also associate weights that reflect the importance of each term as an indicator of the content of the documents to which it is assigned. Thus in designing storage and search strategies, it is reasonable to represent documents by a document-term matrix, where the (i, j)th element of the matrix corresponds to the weight of a term j in a document i. In what follows, we denote this matrix by D, having elements w_{ij}. In other words, each document is represented by a vector composed of term weights. This representation (organization) is referred to as a vector space model. An in-depth description of information retrieval systems that model documents and queries as vectors can be found in Salton [Sa83]. The model provides a basis for understanding the basis of determining term discrimination values.

Raghavan and Wong [Ra86] have critically reviewed the vector space model by using notation that more clearly brings out the problems and challenges that are associated with its usual use. Raghavan and Wong believe that the way in which the vector space model has been introduced and used in IR has suffered difficulties. For example, in order to relax the assumption that terms are pairwise orthogonal, term co-occurence information has been used and some methods of computing term correlations suggest that the rows of D can be viewed as vectors corresponding to the terms, i.e., $t_j = (w_{1j}, w_{2j}, \ldots w_{ij})$. However, Ragahavan and Wong demonstrate that representing documents as rows in D is not consistent with representing terms as the columns of D. The vector space model will be discussed more fully in Sect. 3.

Probabilistic indexing theory states that the best index terms are those that tend to occur in the relevant documents with respect to some query. In using probabilistic methods, a central problem is the identification of a few relevant documents that can form the basis for relevance feedback. Griffiths, Luckhurst, and Willett [Gr86] performed a study of various clustering methods to retrieve groups of similar documents and found that these methods produced large numbers of small clusters containing just pairs of documents. This motivated them to consider the use of

clusters consisting of a document and another with which it is most similar. A comparison of the use of such clusters with conventional best match searches suggests that the two types of search are of comparable effectiveness, but they retrieve noticeably different sets of relevant documents.

2.1.4 Application of Database System Techniques

Database systems generally do not address the needs of information retrieval systems very well. In particular, database fields are usually of fixed length and expect to have one value in a field. Eastman, however, in a review article on database management systems [Ea85], provides a short discussion of text-oriented data models. She mentions such features as the allowance of multiple authors and keywords by extending the relational data model to allow non-first-normal form relations and also provision for arbitrarily long text fields. Use is made of a database system in JANE, however, but only fixed field records are stored with an index file managed by the database system. The documents, themselves, are stored in a direct access file not managed by the database system.

2.2 Automatic Indexing and Retrieval

Although subject analysis is a broader topic than indexing, per se, we include a review here by Schwartz and Eisenmann [Sc86] because of the discussions of indexing that are included. In this regard, they quote Cleverdon [ClC84] as asserting that indexing is an activity whose time has passed, but they think this may become true in the future, and is not the case as of now.

They report that experience with the Library of Congress Subject Headings has led to the belief that precoordinate indexing is inappropriate for online retrieval, although this remains to be proven. Precoordinate indexing is the assignment of a complex index entry (usually a descriptive clause) to represent a concept rather than a number of simple terms which (taken together) represent the concept. They also quote Sievert and Boyce [Si83] who challenge the traditional view of controlled vocabularies as recall devices and show that breaking down multiword descriptors into single-word postings in inverted files enhances precision. Radecki [Ra82] considers the fuzzy-set theory, discussed in the next section, to be the only theoretically justifiable approach to the problem of merging competing models, but he admits that the fuzzy-set model requires weighting methodologies that are not yet developed. In Sect 6., the new JANE system will be used to compare

the retrieval results using automatic indexing with the results using the RSIC precoordinated descriptor system which should help resolve the issue.

The topic of automatic indexing is covered briefly by Borko [Bor82]. He suggests that manual indexing at the input stage is too costly, and suggests several alternatives. One is to produce an inverted list of terms at input time but weight the terms at search time. He also discusses how recall and precision can be improved, and then goes on to raise the question whether recall and precision ratios are the best criteria for evaluating the effectiveness of online retrieval systems. He states that the retrieval process can be fine-tuned to provide any degree of recall and precision, and so new evaluation measures are necessary. Unfortunately, he does not suggest what they are.

As mentioned above, Term Discrimination Values (TDV) have been suggested as an effective means for the selection and weighting of index terms in retrieval systems. Willett (Wi85) reports an algorithm for the calculation of term discrimination values that is sufficiently fast in operation to permit the use of exact values, rather than the approximate values studied in work prior to Willett. This could be an altenative to the method implemented in JANE.

Evidence is presented by Willett to show that the relationship between term discrimination and term frequency is crucially dependent upon the type of inter-document similarity measure that is used for the calculation of the discrimination values. In plots of average rank as a function of term frequency, it is seen that a well-marked minimum is obtained with the cosine coefficient, while the dot product and Euclidean distance result in plots that increase or decrease monotonically as the term frequency increases. He does not attempt to explain this. This suggests that the theory is still not fully understood and testing systems with real collections is necessary.

Recent work by Can and Ozkarahan (Ca87) suggest a more efficient computation of TDVs than previous algorithms. They introduce the concept of Cover Coefficient (CC) which was originally introduced for document clustering puposes. A decoupling coefficient is computed for each document which is a measure of how "decoupled" the document is from the other documents in the collection. The document space density is similar to the overall decoupling coefficient of documents in the following sense. If document descriptions are more distinguishable, then this means that the documents are more decoupled from each other. Thus, the CC concept may be used in computing TDVs which, by definition, depend on the document space density. Further detail of this method is given in Sect. 3.1.3 which describes implementation of the new system. Can and Ozkarahan

also show how TDV and CC concepts can be used for "tunable indexing," a means of regulating the number of document clusters in the collection, and finding optimum weights for indexing.

The original paper on the measurement of term importance, including Term Discrimination Value, was published by Salton, Wu, and Yu [Sa81] in 1981. They develop a sparse vector theory which defines a space density as the average similarity between all distinct document pairs in a collection. The discrimination value now specifies that a good content term is one which decreases the space density when it is assigned to the items of a collection because in that case, the documents are most easily distinguishable from their neighbors. Again, details are deferred to the section on implementation in Sect. 4.

Statistical methods can also be used to identify words in a document that may have superior value in indexing. Harter [Ha75a, Ha75b] has developed a probabilistic approach to automatic keyword indexing. The research was prompted by the observation, made by a number of writers, that non-specialty words, words which possess little value for indexing putposes, tend to be distributed at random in a collection of documents. In contrast, specialty words are not so distributed. In this study, a mixture of two Poisson distributions was examined as a model of specialty word distribution, and a statistical measure intended to identify specialty words was proposed. In the second part of the study, an algorithm defining a measure of indexibility was developed and was found to produce indexes superior to those produced by a term frequency measure which had previously been found to produce the best results.

Dillon and Federhart [Di84] also used statistical methods to locate superior indexing terms. They discuss ways to inprove the quality of retrieval systems that depend on the use of truncated words or quasi-word stems as an indexing vocabulary. Despite inherent weaknesses, the stem, by contrast to longer units like the phrase, remains the only practical method of fully automatic indexing. The problems addressed are the generalizability and stability of discriminant function analysis for selecting good topical terms from terms of relatively high frequency in a database drawn from abstracts of Harris Survey press releases. Results confirm that topical terms can be identified by their statistical properties. Consistently high recall of topical terms under a variety of different conditions inplies persistent underlying properties strong enough to resist changes in test environment.

An interesting article by Dutta and Sinha [Du84] discusses what they call a pragmatic approach to indexing. Although the system cannot be considered "automatic indexing," after a knowledgeable indexer constructs a logical string, the rest of the operations can be handled by a computer. The keywords are arranged in a meaningful sequence and connected by punctuation marks depicting various types of associations. The keywords are rotated to provide access through each significant keyword. The system, which is computer-manipulative, can also be used manually.

Cleveland, Cleveland, and Wise [ClD84] investigated whether full-text indexing was necessary for effective retrieval. They found that by using a particular non-Boolean method as a file structuring and searching technique, full-text indexing is not essential to optimum information retrieval effectiveness. This is encouraging because most collections are not available in full-text form. In fact, the RSIC database has only the title and abstract for automatic indexing.

The importance of ranking documents presented to the user as the result of a search is emphasized by Radecki [Ra85]. He has investigated rigorous methods for incorporating a weighting mechanism into conventional Boolean retrieval systems in order to provide lists of ranked output documents. This has been achieved by developing a formal theory linking Boolean retrieval systems with established probabilistic retrieval procedures applicable in the case of queries as well as documents represented simply by sets of unweighted index terms. Ranking of the retrieved documents is an important feature of the JANE system.

2.3 Fuzzy-Set Applications to Information Retrieval

In their discussion of the problems of combining Boolean query processing with term weighting systems, Salton and McGill [Sa83] state that this approach raises a number of difficult problems, because the normal two-valued logic used to process Boolean queries assumes that a term is either present in or absent from a document or query identification. When the Boolean system is extended to include term weight or importance factors, it becomes necessary to interpret the meaning of compound expressions consisting of weighted terms and Boolean operators. In so doing, one would like to define matching functions that resemble the normal Boolean operations and produce equal output for distinct query formulations that are logically equivalent. It is possible to design retrieval models that furnish unambiguous retrieval output for weighted Boolean queries. The theory of fuzzy sets provides a model in which the weight of term j in document i represents the degree to

which the document is a member of the set of documents indexed by term j. The manipulation of compound Boolean expressions can also be defined in the fuzzy set model.

In a three-volume collection of papers on fuzzy set applications, Buckles and Petry [Buc87] review generalized database and information systems. They state that the issues in information retrieval revolve about determining how well the query terms match keywords, how well the keywords match the document, and the relative importance of the query terms.

Bookstein [Bo85] has published an extensive review of fuzzy-set applications to information retrieval. He points out that although the concept of probability appeared early in the information retrieval literature, the application of fuzzy-set theory and probability theory to the development of optimal information retrieval algorithms is relatively recent, with the few earliest papers appearing only about twelve years ago. Since then, this avenue of research has attracted a considerable number of adherents within the academic community; it remains the source of lively research activity.

Bookstein considered fuzzy requests in a paper published in 1980 [Bo80]. In this paper, he discussed how the fuzzy set concept forms the basis of the theory in which weights are assigned to index terms and logical operations are performed as in conventional Boolean algebra. The properties of such a system are discussed, and it is shown that such systems retain the manipulability of traditional Boolean requests.

In a 1981 paper, Bookstein [Bo81] isolates a number of criteria that should be met by any Boolean system generalized to have a weighting capability. It is proven that only one weighting rule satisfies these conditions, the one normally associated with fuzzy-set theory, and that this weighting scheme satisfies most of the other properties associated with Boolean algebra as well. Probabilistic weighting is then introduced as an alternative approach and the two systems compared. In the limit of zero/one weights, all systems considered converge to traditional Boolean retrieval.

Prade and Testamale [Pr87] recently studied representational issues and reduction using similarity measures for fuzzy relational databases. When several items have sufficiently similar values for each attribute in a relation, the reduction is performed by taking for each attribute the union of similar values.

Buell [Bu85] has pointed pointed out some of the problems of extending the logic of Boolean structures to weighted queries. Certain properties of Boolean structures must be relinquished if relevance weights are to be attached to terms and expressions. The paper of Bärtschi [Ba85] also summarizes these problems and how several researchers have dealt with them. Bookstein has accepted the loss of structure in his suggested system with relevance weights. Buell issues a warning that further thought must be given to the benefits and problems involved in keeping or losing various axioms of Boolean structures. In contrast to Boolean structures, which are thoroughly understood, these new structures proposed by various authors, are still not clearly defined and completely explored, either as mathematical objects or as models for retrieval systems.

In a recent short note, Yager [Ya87] suggests an alternative methodology for handling weighted queries in a fuzzy environment. A problem of considerable interest to designers of retrieval systems based on fuzzy set theory concerns itself with the evaluation of the retrieval status value (RSV) in the situation in which the query terms or the search criteria have weights indicating their importance to the requester. A number of approaches have been suggested for this problem, but Bärtschi [Ba85], as mentioned above, points out some of the difficulties with each of these proposed methods. Yager suggests an alternative methodology for handling weighted queries in a fuzzy environment. His method was adopted for the JANE program developed for this thesis research. The details are given in Sect. 4.

2.4 Artificial Intelligence

The following main roles have been assigned to various artificial intelligence (AI) techniques in information storage and retrieval systems [De83]:

(a) The implementation of natural language interfaces of user-system interaction in database processing.

(b) The introduction of complex text indexing systems, based on available knowledge of the collection domain.

(c) The construction of rules for automatic data entry where formalizations of new data are automatically constructed in terms of specified meaning representations that can then be used to update the stored knowledge base.

(d) The use of active memory techniques capable of enforcing (semantic) constraints on the elements of the stored knowledge base, and of making the inferences that are necessary to supply new elements and new relationships derived from the already available know-how.

(e) The introduction of expert system techniques which guide the query analysis and search and retrieval processes, based on knowledge about user needs and about the effects of various search techniques on the retrieval output.

Thornburg [Th86] has recently discussed the possible roles of expert systems in information storage and retrieval and explored the issues in the development of such systems. She identifies some of the main issues of dealing with expert systems in this context, discusses some individual systems, and reviews some needs and issues not yet addressed in actual applications. Thornburg also suggests that machine learning techniques such as conceptual clustering [Mi83] might be useful in studying the inverted index of an online vendor's system, as for example to provide understanding of the subject coverage of the underlying databases.

Salton [Sa86] has also examined the prospects of using artificial intelligence methods to improve information retrieval. He indicates that experimental evidence indicates that substantial difficulties arise in obtaining effective complex text identifiers, as opposed to single terms, that actually help in retrieval. He reaches the conclusion that AI methods are likely to be even more difficult to apply to normal document environments than the conventional text analysis methodologies.

In another application of knowledge-based systems, Humphrey and Miller [Hu87] discuss computer-assisted indexing of the medical literature. The project entails two areas, indexing for information retrieval and knowledge representation.

The role and contributions on natural-language processing information retrieval has been examined by Doszkocs [Do86] in the context of large operational systems and services. State-of-the-art information retrieval systems are found to combine the functional capabilities of the conventional inverted file, Boolean logic, term adjacency approach, that are commonly employed by commercial search services, with statistical-combinatoric techniques and natural language processing methods and tools borrowed from artificial intelligence.

Intermediary expert systems are designed to mediate between end-users and complex information retrieval systems. However, since most of these expert systems are based on text analysis rather than on models of human searching, they cannot process request-related criteria, such as precision and recall requirements. Fidel [Fi86] discusses research in which analysis of the searching

behavior of human intermediaries revealed a routine for the selection of search keys along a decision tree. Examples of decision rules demonstrate that these rules can be automated to significantly enhance the adaptability of intermediary expert systems.

In two recent papers, Biswas et al. [Bi87a, Bi87b] describe a system with a natural language interface based on fuzzy set techniques to handle the uncertainty inherent in natural-language semantics. The user query is considered to be triple which consists of concepts and operators to link those concepts, the publication period desired, and the number of documents to be retrieved. The output of the interface is then passed to a knowledge-based retrieval mechanism. The retrieval mechanism is a two-step procedure. In the first step, a list of documents are obtained from the document database, and then an evidence-combination scheme is used to compute the degree of support between the query and individual documents. The second step uses a ranking procedure to obtain a final degree of support associated with one or more parts of the query. The end result is a set of document citations presented to the user in ranked order in response to the information request.

3. ALGORITHMS ADOPTED FOR JANE SYSTEM

3.1 Algorithms Adopted for Automatic Indexing

The algorithms implemented in this study are based on the usual assumptions that using single term frequency in a document and increasing the resolving power of the assigned terms are the most effective bases for automatic indexing [Sa83].

Theoretical developments in the theory of information retrieval (IR) have offered a number of alternatives to traditional indexing which may be described as assigning descriptive terms to each document which together indicate the subjects treated by the document. In assigning terms to each document, the indexer builds a vocabulary of terms, the indexing language. Each term is associated with one or more documents, and thus there is a set of documents which are associated with each term. This leads to the concept of a document-term matrix in which the elements, 0 or 1, indicate whether the document is indexed by the term. It is also possible to assign other values, or weights, for the elements, which may be interpreted according to concepts developed within probability theory or fuzzy-set theory. Fuzzy-set theory is an attempt to generalize traditional set theory by permitting partial membership in a set. In an IR context, this means recognizing different degrees to which an index term can apply to a document.

3.1.1 Document-Term Assignment

Within the IR context, a fuzzy set of documents is associated with each term; that is, for a given term, a membership function is defined that expresses the system's judgment as to what extent each document is about that term. Thus, in contrast to traditional Boolean systems, a continuum exists between those documents that are not at all relevant to a term and those documents that are centrally relevant to the term. In this sense the boundary that separates documents that are relevant to an index term from those that are not relevant can be said to be fuzzy. The set of documents relevant to an index term is created during the indexing process, that is, as terms are assigned to documents the membership in the set for each term is defined. In this manner, once all documents have been indexed, a membership function has in effect been defined for each document indicating the extent to which an indexing term is associated with a document.

Conversely, each term is associated with a fuzzy set of documents that are relevant to the given term.

We denote the document collection containing m documents d_i to be indexed by C,

$$C = \{d_i | 1 \leq i \leq m\}.$$

The set of n index terms, i.e., the indexing vocabulary for the description of these m documents, is

$$T = \{t_j | 1 \leq j \leq n\}.$$

The description of the document collection, D, is a matrix of size $m \times n$. An individual entry of D, w_{ij} $(1 \leq i \leq m,\ 1 \leq j \leq n)$, indicates the relative importance or weight of a term t_j in a document d_i, where $w_{ij} \in W$ and W is the set of possible values.

In traditional systems $W = \{0, 1\}$, which means that index terms not assigned to a particular document can be interpreted, within this formalism, as having been assigned a weight of zero. This permits us to consider each term as having had a weight assigned to it. In the JANE system, we accept all words in the title and abstract as indexing terms, except words which appear in a file called the "stop list." These are words of poor indexing value, such as articles (the, a, an) or other trivial words. The words in the stop list are generally the very-high frequency words [Sa83]. The term discrimination value (TDV), as described in Sect. 3.1.3 is also used to determine terms for the stop list. We define a word as a sequence of alphabetic characters, truncated, if necessary, to eight letters to form a stem. The stem, with all characters in upper case, is taken to be the term. A preliminary file of 500 documents was processed to help determine a stop list which was used when the present database of approximately 4800 documents was processed.

3.1.2 Document-Term Frequency Assignment

In a fuzzy or probabilistic information system, a term weight within a unit interval [0,1] is determined when a term is assigned to a document. The weight value is assigned according to a perceived importance or degree of relevance of the term in the document. This value is frequently determined on the basis of the frequency of occurrence of the term in the document [Sa83]. In the JANE system, we determine the weight value of a term for a given document from its frequency of occurrence in the title and abstract.

The first step in creating the INFOS II database records is the extraction of terms, t_j, from the title and the abstract of each document. The program that extracts the terms creates a file of document-term pairs. This file is then processed to determine the frequencies of the terms in each document, which are designated f_{ij}. Because terms from the title are more narrowly focused than terms from the abstract and the abstracts vary greatly in length, the values are reduced for long abstracts to compensate for the increased probability of occurrence independent of its importance.

The modified frequency of a particular term t_j in a document d_i is computed as follows:

$$f_{ij} = \text{FREQ}(t_j, \text{title}_i) + \text{FREQ}(t_j, \text{abstract}_i) \times F,$$

where

$\text{FREQ}(t_j, \text{title}_i/\text{abstract}_i)$ = number of occurrences of term in title/abstract of document,

$$F = 1 \quad \text{for } a \leq 10,$$

$$F = \frac{1}{1 + \log_2(a) - \log_2(10)} \quad \text{otherwise,}$$

and a = number of terms in abstract.

The use of the function F is an attempt to compensate for the wide variation in the length of abstracts as mentioned above.

3.1.3 Term Discrimination Value

An indexing theory proposed by Salton and his coworkers [Sa83] uses the term discrimination value (TDV) concept. In this theory, an optimum indexing vocabulary in obtained according to the significance of terms. For computing term significance, several weighting measures have been proposed, such as term frequency, signal-to-noise ratio, variance dynamic information value, and discrimination value. The TDV values are used to modify the term weights with the expectation that measures of retrieval performance, such as recall and precision, will be improved.

Recall is defined as the fraction of the relevant documents in the whole collection that are retrieved in response to the query, i.e.,

$$\text{recall} = \frac{\text{number of relevant documents retrieved in query}}{\text{number of relevant documents in the collection}}.$$

Except for very small collections, the denominator is impossible to determine. In the test comparisons of the new and old systems discussed in Sect. 6, the measure of performance is taken to be number of relevant documents retrieved. This is equivalent to recall, because the systems are presented with equivalent queries and search the same collection.

Precision is defined as the fraction of documents retrieved by a query that are relevant to the query, i.e.,

$$\text{precision} = \frac{\text{number of relevant documents retrieved in query}}{\text{number of documents retrieved in query}}.$$

Queries can usually be reformulated by a searcher to improve precision, but this is likely to reduce recall. On the other hand, a reformulated query may have the opposite effects. A balance is generally sought.

A systems designer must consider optimizing both recall and precision that the system will exhibit. The recall is proportional to the frequency of index term occurrences in the documents, but high frequency words, providing high recall, leads to poor precision. To improve precision, the frequencies can be modified to emphasize the terms which distinguish documents from each other. This may be done by using the term discrimination value as discussed below. Therefore, the weights, w_{ij}, for the document-term matrix are a function of both the frequency in the document and the TDV of the term.

In a document space, TDV is used to measure an index term's t_j contribution to how well documents are separated (distinguished) from each other. The separation of a collection of documents d_i can be measured by looking at the document space density, Q, that can be computed as the average similarity of documents,

$$Q = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S(d_i, d_j) \qquad (0 \leq Q \leq 1 \quad \text{for } 0 \leq S \leq 1). \tag{1}$$

$S(d_i, d_j)$ is the similarity between documents d_i and d_j, which can be calculated according to the cosine coefficient [Sa83] as

$$S(d_i, d_j) = \frac{\sum_{k=1}^{n} f_{ik} \times f_{jk}}{[\sum_{k=1}^{n} (f_{ik})^2 \times \sum_{k=1}^{n} (f_{jk})^2]^{1/2}}. \tag{2}$$

Deletion of a term will change the indexing vocabulary and hence the description of the documents and the document space density. Let the average space density without a term t_j be denoted Q_j. Then the term discrimination value of the term t_j is given by

$$TDV_j = Q_j - Q. \tag{3}$$

The TDV has the following properties:

(a) TDV is positive for a good discriminator t_j (i.e., the term t_j makes the documents more distinguishable from each other),

(b) TDV is approximately 0 for an indifferent term t_j (i.e., the assignment of the term t_j to documents does not affect the separation of documents), and

(c) TDV is negative for a poor discriminator t_j (i.e., term t_j makes the documents less disinguishable from each other).

The TDV values then can be used to modify the weights of the document-term matrix and optimize the indexing vocabulary.

If the formula (1) is used to compute TDV values, it can be very costly. Hence approximate methods have been devised, involving calculating the centroid (an average of the document vectors), and improved algorithms have been suggested to compute the exact TDV [Wi85]. Another approach, using the cover coefficient (CC) concept, has been introduced by Can and Ozkarahan

[Ca87]. This concept, originally introduced for document clustering purposes, defines a document-document matrix C onto which the document-term matrix (D) is mapped. The elements c_{ij} of the matrix C are

$$c_{ij} = \alpha_i \times \sum_{k=1}^{n} f_{ik} \times f_{jk} \times \beta_k \qquad (1 \leq i \leq m,\ 1 \leq j \leq m), \tag{4}$$

where α_i and β_k are the reciprocals of the row$_i$ and column$_k$ sums of the D matrix elements as shown below:

$$\alpha_i = \frac{1}{\sum_{j=1}^{n} f_{ij}}, \qquad (1 \leq i \leq m), \tag{5}$$

$$\beta_k = \frac{1}{\sum_{j=1}^{m} f_{jk}}, \qquad (1 \leq k \leq n). \tag{6}$$

Each entry in the matrix C indicates the extent with which a document$_i$ (d_i) is covered by a document$_j$ (d_j). A diagonal entry of the matrix C indicates the extent with which d_i is covered by itself and is called the uniqueness or decoupling coefficient δ_i of d_i. This quantity is computed as follows:

$$\delta_i = c_{ii} = \alpha_i \times \sum_{k=1}^{n} f_{ik}^2 \times \beta_k, \qquad (1 \leq i \leq m). \tag{7}$$

The sum, over the collection, of the decoupling coefficients, δ_i, is taken to be the number of clusters into which the documents can be separated. This sum is computed with and without a particular term t_j. In a manner analogous to the computation of Q, as discussed above, the TDV is taken to be the difference in the number of clusters before and after deleting the term t_j. The value of TDV_j is then computed from:

$$TDV_j = \sum_{i=1}^{f_j} \left[\delta_i - \alpha_i^j \times \left(\frac{\delta_i}{\alpha_i} - f_{ij}^2 \times \beta_j \right) \right], \tag{8}$$

where f_j is the number of documents referenced by term t_j and superscript j notes the absence of term t_j, that is,

$$\alpha_i^j = \left(\sum_{k=1}^{n} f_{ik} \right)^{-1} = (\alpha_i^{-1} - f_{ij})^{-1}, \quad k \neq j. \tag{9}$$

Can and Ozkarahan [Ca87] present a simple algorithm for the computation of TDV, and this was adopted for this study. It was indeed found to be a practical calculation for the RSIC minicomputer. The algorithm is as follows:

for d_i $(1 \leq i \leq m)$
  compute sum of row$_i$, $S_i = \sum_{j=1}^{n} f_{ij}$
  compute α_i from Eq. 5
endfor

for t_j $(1 \leq j \leq n)$
  compute β_j from Eq. 6
endfor
for d_i $(1 \leq i \leq m)$
  compute δ_i from Eq. 7
endfor
for t_j $(1 \leq j \leq n)$
  $\mathrm{TDV}_j = 0$
  for d_i $(1 \leq i \leq m$, and $f_{ij} > 0)$
    $\alpha_i^j = (S_i - f_{ij})^{-1}$
    $\mathrm{TDV}_j = \mathrm{TDV}_j + [\delta_i - \alpha_i^j \times (\delta_i/\alpha_i - f_{ij}^2 \times \beta_j)]$
  endfor
endfor

In the usual application of TDV to optimize the vocabulary, the term weight, according to the resolving power of the term, is obtained as:

$$w_{ij} = f_{ij} \cdot TDV_j, \tag{10}$$

where f_{ij} is the frequency.

If this approach is used, all terms with a negative TDV value must be discarded so that the weight can remain positive. In our case, it is observed that some terms have a very large negative value but must be retained so that the collection can be subdivided into large sets. For example, most of the documents should be indexed by neutron or gamma or both, and a user will probably want to limit a search to one or the other. Therefore, a different expression relating the term discrimination value of term t_j, TDV_j, and f_{ij}, was formulated:

$$w_{ij} = f_{ij} \cdot e^{TDV_j}. \tag{11}$$

The values of w_{ij} are taken as the final weights for the document-term matrix.

Using the implementation (11), negative values reduce the weight, but can be retained in the vocabulary. Positive values increase the weight as is desired. For the vocabulary developed from the RSIC database, the TDV values varied from about -10 to 1. The values of the weight w_{ij} varied from 0 to about 5.4. Therefore, strictly speaking, we are not dealing with fuzzy sets, but

with sets that are essentially similar in the sense that the higher the value of the weight, the more descriptive the corresponding term is.

For a "term" t_j that is an author's name in a document d_i, the weight w_{ij} is taken to be 1 for any document d_i.

3.2 Algorithms Adopted in Document Retrieval

Zadeh [Za65] defined fuzzy versions of the conventional set operations. If a document is in a set A to a degree a, and in a set B to a degree b, where a is greater than b, then it is in:

- the union of A and B to the degree a,
- the intersection of A and B to the degree b,
- the complement of A (or of B) to the degree $1 - a$ (or degree $1 - b$), and in
- the difference of A and B to the degree $a - b$.

We shall define a query as taking the form of a term or author's name or a Boolean expression of terms and names. In response to a query, the system retrieves a set of documents which are associated with the query to a degree specified by the term weight w_{ij}. As additional queries are presented to the system, additional sets are retrieved, and these can be combined, with the query taking the form of Boolean expressions of set identifiers. For example, if a disjunction (OR) of a set identified as Set_1 and a set identified as Set_2 is desired, the form of the query would be:

$$\text{Set}_1 \vee \text{Set}_2.$$

We may also consider a form of query in which the user assigns an importance or weight to the elements of the of the query. For example, in a conjunctive (AND) query, the user might assign an importance q_1 to the set Set_1 and importance q_2 to the set Set_2, remembering that each set is associated with a particular term or group of terms. He could then form the query:

$$\text{Set}_1(q_1) \wedge \text{Set}_2(q_2).$$

Given the above definitions of fuzzy sets and set operations, the retrieval process proceeds as follows: sets of documents are retrieved in response to requests, and the sets are combined by set operations as in the traditional case (conventional Boolean retrieval). Again, a set of documents is the result of a search. Operationally, the degree to which the documents in the retrieved sets satisfy the query (are members of the retrieval set) are used to rank the documents for presentation to the user, with the most relevant documents (those with the largest membership or "retrieval status" values) being presented first.

The Retrieval Status Value (RSV), then, is w_{ij} for the documents retrieved when the query Q is a term and $R(d)$ when the new set is the result of combining other sets. That is,

$$\begin{aligned} \mathrm{RSV}(d_i) &= w_{ij} \qquad \text{if } Q \equiv t_j, \\ &= R(d_i) \quad \text{if } Q \equiv \mathrm{Set}_1(q_1) * \mathrm{Set}_2(q_2), \end{aligned}$$

where $*$ can be $\wedge$, $\vee$, or NOT.

A problem of considerable interest to designers of fuzzy-set retrieval systems concerns itself with the evaluation of the RSV in the situation in which the query terms of the search criteria have weights indicating their importance to the requester as mentioned above. A number of approaches have been suggested for this problem, but Bärtschi [Ba85] points out some of the difficulties with each of these proposed methods. Yager [Ya87], in a recent paper, proposes an alternative methodology for handling weighted queries which is simple to implement and intuitively appealing. His method was adopted for this study.

We have implemented the Yager algorithms in the JANE program in the retrieval option which combines two sets. Each set is a list of documents and their associated weights (retrieval status values). A set is associated with a term or author, or it has been created when two other sets have been combined, in which case the resulting set may be regarded as being associated with prior "queries." Methods of computing the RVS value are given below for all combination operations (union, intersection, and difference).

The Yager method for a conjunctive (AND) query of the form:

$$\mathrm{Set}_1(q_1) \wedge \mathrm{Set}_2(q_2) \qquad (q_1, q_2 \in [0,1]),$$

proceeds as follows:

The Retrieval Status Value (resulting weight) of a document d is

$$R(d) = \min[I(q_i, C_i(d))] \qquad (i = 1, 2), \tag{12}$$

where $C_i(d)$ is the membership value (weight) of document d in the retrieval sets Set_i, q_i is the degree of importance of satisfying a criterion C_i, and I is the function relating importance and satisfaction. $C_i(d)$ is the degree of satisfaction of the document d to criterion C_i.

Yager suggests the following form for I based on his research:

$$I(a, b) = \max(1 - a, b) \qquad (a, b \in [0, 1]).$$

In our implementation $a = q_i$, and $b = C_i(d)$. We cannot guarantee that $b \leq 1$ because w_{ij} values vary between 0 and about 5.4. However, only a negligible percentage exceed 1. Therefore, we can modify I to

$$I(a, b) = \min[1, \max(1 - a, b)] \tag{13}$$

to normalize the value of I. Substituting (13) into (12), we have:

$$R(d) = \min[1, \max(1 - q_1, C_1(d)), \max(1 - q_2, C_2(d))], \tag{14}$$

which is implemented in the JANE retrieval program.

We may note that if a criterion is of no importance ($q_i = 0$), then $I(q_i, C_i(d)) = 1$, hence all documents in the retrieval set are retrieved. Furthermore, when $q_i = 1, I(q_i, C_i(d)) = \min[1, C_i(d)]$, which is nearly always $C_i(d)$. Thus the pure (Boolean) conjunctive query, in which query weights are not considered, is a special case of the general conjunctive query in which search requests which *must* be satisfied are given a weight of 1 and those of lesser interest are given a weight close to 0.

Since $q_1 > q_2$ implies $[(1 - q_2) \vee a] \geq [(1 - q_1 \vee a]$, and since smaller valued terms are more influential in the evaluation of the min operation, we see that the more important the criterion, the more influential it is in the determination of the RSV ($R(d)$).

In the disjunctive (OR) query of the form:

$$\text{Set}_1(q_1) \vee \text{Set}_2(q_2),$$

where q_i is the importance of the Set_i in the disjunctive query, the resulting RSV, or weight, of a document d in the two sets being combined is:

$$R(d) = \max[U(q_i, C_i(d))] \qquad (i = 1, 2), \tag{15}$$

where, as Yager suggests,

$$U(a, b) = \min(a, b). \tag{16}$$

Thus,

$$R(d) = \max[\min(q_1, C_1(d)), \min(q_2, C_2(d))], \tag{17}$$

which is implemented in the JANE program. Here, since $q_i \in [0, 1]$ and minimum of q_i and $C_i(d)$ is taken, U is normalized.

We note that since the max operation is more affected by larger-valued arguments, the larger $U(a, b)$ is, the greater role it plays. Thus, the larger q_i is, the more significant role the corresponding criterion C_i plays in the clause. If $q_i = 0$ (since $0 \wedge C_i(d) = 0$ and $\max[b, 0] = b$), then the elements in the set S_i are not retrieved.

In the set difference (NOT) query of the form (without query weights):

$$\text{Set}_1 \; - \; \text{Set}_2, \tag{18}$$

the RSV is:

$$R(d) = \max[0, C_1(d) - C_2(d)]. \tag{19}$$

The above algorithm was used as a plausible means to complete the operations that were needed since no algorithm for this operation was found in the literature. In the case of a small document weight in Set_2, the resulting weight is only slightly reduced. In the case of a larger or equal weight of the document in Set_2, as compared to the document in Set_1, the document is eliminated. These results are intuitively correct. Query weighting has not been implemented for the set difference (NOT) operation.

Usually, only a portion of the retrieved documents whose RSV satisfies a given *threshold of acceptance* (i.e., $RSV(d) \geq T$), are presented to the user. The threshold of acceptance within the fuzzy-set based retrieval system is a number in the unit interval [0,1]. Hence, the size of the retrieval set can be controlled by choosing an appropriate value of the threshold of acceptance. A value of 1 is equivalent to Boolean retrieval (documents are retrieved where $w_{ij} \geq 1$). A value of $T = 0$ would result in all documents being retrieved. In our system, however, only indexed documents can be retrieved (i.e, $w_{ij} > 0$), even though the user sets the threshold to 0.

Documents in the retrieval set are sorted in the decreasing order of their Retrieval Status Values, when presented to the user. The underlying assumption is that documents with higher RSV are more relevant to the query.

4. DEVELOPMENT OF THE NEW SYSTEM

When it was decided to shut down RECON, the Department of Energy requested bids from commercial vendors to supply database search services. From the response of potential vendors, it was obvious that the small client community of services such as RSIC preclude commercial viability, and the costs would be too great to store and maintain the database on commercial systems. This also proved true in regard to the computer systems operated by DOE. Thus it is necessary for RSIC to develop its own system.

4.1 Requirements

As mentioned in Sect. 1, RSIC needs a new system to replace the current ADSEP/ORLOOK combination. Database systems are available on the ORNL mainframes, but database systems are not particularly suitable for information retrieval. Furthermore, use of the central system is costly and cumbersome as compared to the use of the RSIC minicomputer.

In designing the new system, several desirable features have been incorporated. For example, *automatic indexing* has been implemented, and if further testing shows the system to be practical, great savings in nuclear engineers' time will be achieved. It will also be of great benefit if term weights and fuzzy set logic do, indeed, improve retrieval results and the allow presentation of results to the user in the decreasing order of query satisfaction. The number of documents retrieved can be controlled by the user, and their presentation is in predicted relevance order. That is, the system determines the RSV for each document as discussed in the previous section, and these values are the system's estimation of the degree of relevance to the query.

Domain knowledge relevant to the shielding database has been incorporated in the system through a thesaurus. This is important in that it helps the user to automatically increase the scope of a search using equivalent terms. This is especially of value to novice users who may not be thoroughly familiar with the specialized terminology used in the fields of shielding and ionizing radiation transport.

Finally, an attempt has been made to incorporate user preferences in system operation, making it "*user friendly*." Most of the users would be using the system infrequently, so a series of menus

are used to guide the search. Each menu gives a brief explanations of the options, and frequently defaults can be taken to minimize input. A special option is available which can display a section of the term or author index. This option allows the user to select keys (terms or authors) which are closely related in spelling to the entered search key. Another option allows the use of "generic" keys, whereby all keys are selected that begin with the entered key. This is particularly useful for including plural forms or misspelled terms or authors in the database. Since the user interface is locally programmed, it will be easy to incorporate additional features as experience and user feedback accumulate.

4.2 Resources

In contrast to the use of outside vendors, the RSIC Data General (DG) Eclipse MV/4000 computer can be used with little or no additional cost. There are two disks, of approximately 350 MB each, shared by all users. If necessary, additional disk space can be purchased.

The software available includes a Fortran 77 compiler and the DG INFOS II database system [IN85]. The two are used together. Fortran application programs provide a user interface that calls subroutines which pass data to and from the INFOS II server. The INFOS II server manages the tree structure of keys and INFOS II records which contain pointers to the bibliographic records in a direct access file. The SORT/MERGE utility [SO85] is also available to insert and retrieve data from the INFOS II files in a batch mode.

4.3 Implementation Overview

4.3.1 System Organization

The database is organized as a direct access linear file for rapid record lookup. The Data General INFOS II system is used to store an index of search keys and a database of records containing terms, weights, and pointers to documents in the linear direct access file. It also is to provide rapid access to these records. The INFOS II files are equivalent to an inverted file containing the document-term matrix which contains the weight for each indexing term and document. With this organization, no space is occupied by empty elements of the matrix.

As mentioned above, A Fortran program acts as a user interface to accept queries, construct sets as results to the queries, perform fuzzy-set operations on the sets, and present results to the user. The Fortran program contains INFOS II interface routines which provide a means to interact with the INFOS II server to access the INFOS II files. More information on INFOS II, and utilities used with INFOS II, is given below. The Fortran program is called "JANE" in honor of a person long associated with maintenance and searching of the RSIC database. The system organization is schematically shown in Fig. 4.

4.3.2 Data Structures

The RSI data base has been converted to linear direct-access files on the Eclipse disk. The records, or documents, are actually document surrogates containing bibliographic fields of varying lengths. Depending on the status of a particular record, it may or may not have an abstract and indexing fields. As mentioned earlier, the documents, usually without an abstract or indexing fields, are entered into the data base when the document becomes known to the RSIC staff. At a later time, the abstract and indexing fields are added to the record.

In the present database, the indexing information consists of RSIC subject category codes (a variable number of them), each with a three-level weight (emphasis), an indicator code, and keywords. The 'indicator' code carries such information as to whether the document is an exposition of a theory or merely uses it, or whether the data presented has been compared with other research results. The keywords correspond to the assigned subject category codes, but are given without weights. The keyword field was added to the record when the database was converted for RECON use because most searchers were external to RSIC, and they were more used to using keywords as search keys. The keywords are generated automatically from the subject category codes when the abstract and category codes become available for the document.

For this study, the indexing information is ignored in the construction of the document-term matrix, and only words from the titles and abstracts are used. This approach was chosen so that a comparison could be made of the results of automatic indexing, as implemented in this thesis research, with ORLOOK searches using the manually assigned category codes and keywords. The results of a number of comparison searches are given in Sect. 6.

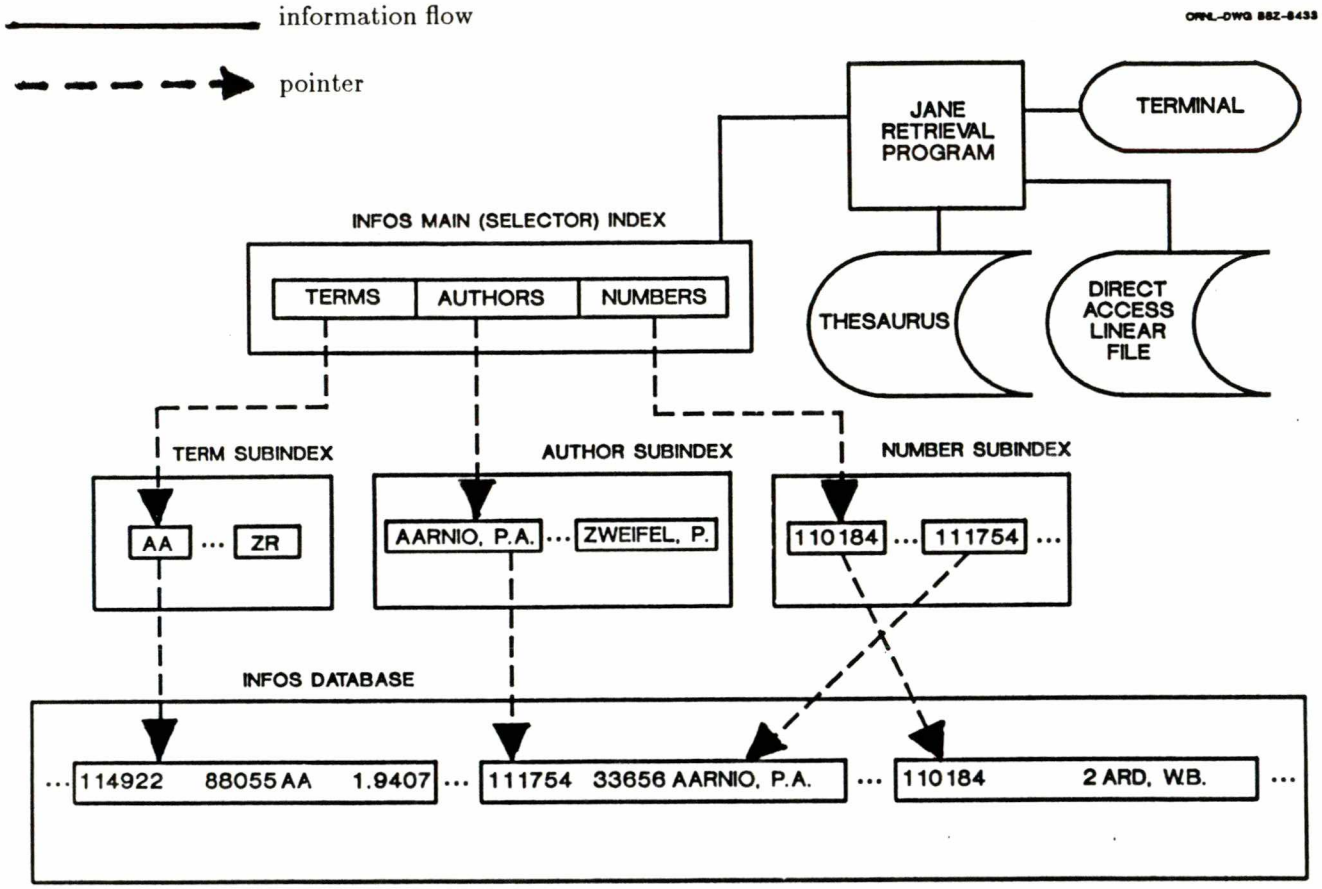


Figure 4. JANE System Organization

4.3.3 Conversion to the Eclipse

The first step in converting the ADSEP files on the ORNL IBM mainframe to a new system is to export the data from the internal ADSEP structure into a text format with labeled fields. This is done by running an ADSEP utility program. The output is written to tape for transfer to the RSIC computer. The tape files are read by the Eclipse and placed in disk storage as sequential files. Unfortunately, the utility writes 80-character records without regard to where words break. Therefore, a program had to be written to process the file and join words which were broken at column 80.

Recall from Sect. 4.3.1 that the overall organization of the data for the system is in two parts, the main file (titles, abstracts, etc.) and the INFOS II database used to provide rapid access to the main file. The ADSEP records (documents) are stored sequentially in a so-called "linear" file. For rapid retrieval, the linear file must be made a Fortran "direct access" file with each 80-character record directly accessible a record number. Therefore, the next step in the process of conversion to the new system is to convert the file from the sequential to the direct access organization. After the file has been converted, a Fortran program is used to create a header record for each document. This document header record contains the direct access record number of the beginning of the document and record numbers of important fields, such as title and author. These record numbers, or pointers, are used later in the process of creating the INFOS II records.

4.3.4 Development of the INFOS II Files

The INFOS II database file contains records with search terms and the corresponding pointers to the direct access linear file or files where the information relevant to a given search key is stored. The INFOS II file is logically equivalant to a fast-access inverted file containing the document-term matrix. Each INFOS II record contains a document accession number, a direct access record number (linear file pointer), and either a search term with weight or an author name. The accession number, sometimes called ADSEP number, is a unique number associated with the document assigned by the RSIC staff to identify the document, and is used in all stages of processing. The record structure is similar to the following:

114922 88055AL .19407

111754 111754AARNIO, P.A.

The 6-character accession number is given in the first field, and is followed by an 8-character field giving the direct access record number. The next fields are either an 8-character term and 8-character weight in decimal form or a 30-character author name.

Another structure, the INFOS II index file, contains search keys stored in a tree structure. Each key points to an INFOS II record. At present, there are three types of keys: term, author, and accession (document) number, and retrieval can be accomplished through access by these keys.

A series of programs were developed to process the linear file to produce the INFOS II records. The weight for each document-term pair to be stored in a record is computed by implementing algorithms described in Sect. 3. From the frequency of occurrence of terms in the title and abstract and the Term Discrimination Value (TDV) as suggested by Can and Ozkarahan [Ca87], they determine the weight for each document-term pair. A list of these programs is given in Appendix B. A stop list of trivial terms was developed, using a preliminary database, to discard terms of little retrieval value.

The first step in creating an INFOS II database is to use the ICREATE utility, a part of the INFOS II system, to create an empty structure. The DG SORT/MERGE utility is then used to insert the records into the INFOS II database. The records do not need to be sorted previously because the INFOS II system maintains the keys in an ordered tree structure. The SORT/MERGE utility was also used several times to sort various intermediate files by document accession number or by term while computing TDV values.

4.3.5 The JANE Retrieval Program

The JANE retrieval program is written in Fortran 77 for the Data General Eclipse MV/family of computers which use the AOS/VS operating system. After compiling the various subroutines of the program, the Data General linkage editor is used to create the executable program, and the necessary INFOS II interface subroutines are included. The program is interactive and uses menus to guide the user. Additional details, including sample menu displays are presented in Sect. 5.

The main menu displays various choices. These are:

1. Retrieve a set using term, author, or number.

The user is prompted for a single term, author, or accession (document) number. In the case of a term, he is also prompted for a retrieval weight threshold (the 'weights' for authors and accession numbers are 1). If the search key entered for a term or author ends in '*', a generic search is performed, i.e., all terms or authors in the index that begin with the entered key are used to create the set. In the case of a search for a term, the user is also asked if a search for synonyms is desired. If so, a thesaurus file is consulted to identify the synonyms, if any. The result of a query using the defined search key is a set which is identified by a number after displaying the menu again automatically upon completion of the search.

2. Retrieve a set using a display of keys.

The user is prompted for a term, author, or accession number. In this case, the program displays 20 keys from the INFOS II index, starting with the first one that begins with the entered key, and the number of documents assigned to that key. The user then is prompted to select a range of keys for the search. He can individually accept or reject keys in the range. The user can scroll another 20 keys, if he desires. In the case that the key is a term, the user is requested to enter a threshold weight for each term selected.

The advantage of this search option is that the user may not know exactly what keys are available for search, or he may not know how an author's name is spelled. The resulting set is a union of the sets retrieved by the individual selected keys. The resulting set is identified when the main menu is again displayed after the sets are combined.

3. Combine sets.

This allows the user to combine the previously created sets by using the logical operators AND, OR, or NOT. The user can also enter query weights indicating the importance of each set to the query. These query weights affect the weights (retrieval status values) of the resulting set as described in Sect. 3.2 on document retrieval.

A new set is created as a result of combining previously retrieved sets and is identified following the menu display.

4. Retrieve a set using a complex query.

This allows the user to create a query that is a Boolean expression built from terms (without query weights) and logical operators AND, OR, and NOT (represented as $*$, $+$, and $-$ respectively), and parentheses.

5. Print records.

This menu selection allows the user to create an output file to print a particular set retrieved. A second menu allows the user to select from the various fields in the RSI record, such as title, authors, or abstract. The user can also restrict the number of citations retrieved. The retrieved documents are presented in descending order by Retrieval Status Value (RSV), i.e., retrieval satisfaction. The retrieved documents are displayed on the terminal screen in addition to being written to a disk file for later printing or other manipulation. It is only at this stage that the linear file is opened to access the RSI records.

6. Exit the program.

A more detailed description of the use of the JANE retrieval program, along with sample displays of the JANE menus, is given in Sect. 5.

5. JANE SYSTEM

5.1 Overview

The JANE system was developed as a demonstration of automatic indexing techniques and retrieval of literature citations and abstracts from the literature collected by the Radiation Shielding Information Center (RSIC). More than 16 000 citations are stored as computer-readable files on the ORNL IBM 3033 mainframe system. A sample set of records, representing individual documents, was transferred to the RSIC Data General Eclipse MV/4000 computer for use in the development of this system.

The system consists of two major parts. The first is a set of programs and files that are used to process the source data and extract the information needed for retrieval. The extracted information is used by automatic indexing programs to produce records and keys which are entered into index and database files structured for fast look-up by the Data General INFOS II database system. Details are given in an Oak Ridge National Laboratory report [Tr88].

The second part is the JANE retrieval program which provides a user interface and which calls the INFOS II server (a process normally running on the RSIC Eclipse computer) for retrieval of the pointers necessary to locate the retrieved records from the linear file. The linear file contains the citations, abstracts, and other data fields. The JANE program also allows the user to combine the retrieved sets by using Boolean logic operators and to print the sets as desired. The JANE program structure is briefly described in Appendix A.

In the current version of JANE, there are three ways to retrieve: by terms (words), by author, and by the accession number. The terms are stems of up to eight alphabetic characters. Documents are indexed by terms and their weights (importance) in a document. In retrieving, a user may specify a threshold weight. The user can take the default value of 0, which retrieves all documents indexed by the term, or he reduce the set retrieved by specifing a higher value such as 0.5. Experimentation is usually necessary. The INFOS II option of generic searching of keys allows the retrieval of records whose keys begin with the characters supplied in the query.

5.2 User-Program Dialog

The user starts the program with the command JANE, which invokes a Command Line Interpreter macro. This macro sets the output list file to be JANE.OUT which will contain the results of the retrieval session. The user is reminded of this at the end of the session.

The user will see a menu on the screen offering a choice of 6 options: (1) Retrieve a set using term, author, or number, (2) Retrieve a set using a display of keys, (3) Combine sets, (4) Retrieve a set using a complex query, (5) Print records, and (6) Exit the program. After completing any option, except the last one, the main menu is displayed again, followed by information on the previously retrieved sets: the number of items in each set and the query which created the set.

If Option 1 is chosen, the user is then asked whether the search key is to be a term, author, or accession number to create a set of the retrieved documents. A term query may contain up to eight letters and an author query up to 30 characters. Upper or lower case is accepted; there is no case distinction. For a generic search, an asterisk is appended to the entered key. In such a search, all documents are retrieved with keys that match the entered key up to the number of characters entered. For example, all terms that start with A will be retrieved if the query term consists of "A*." The longer the key in the query, the more specific the retrieval is. In the case of a term, the user is asked to supply a weight threshold. The threshold weight may be keyed with or without a decimal point but should contain no blanks. The default weight is 0, meaning any weight greater than 0 will be accepted. The user may enter a new line (NL) only to accept this default value. As mentioned in Sect. 3.1.3, the value of w_{ij} can be as large as 5.4, although only a few percent are >1. Hence query weights range between 0 and 1. An accession number should be 8 numerals. A display of the main menu with a choice of Option 1 is shown in Fig. 5 in which the user entries are shown in **boldface**.

Option 2 allows the user to select terms, authors, or accession numbers from a display of 20 available keys. He or she can select a range from the display which shows the number of postings to each key. After selecting a range of keys, the user is asked to verify that he wants to use each key in the range. The final set is a union of the sets retrieved by using the selected keys. The display can be scrolled to display another 20 keys by entering any non-numeric character. If the

```
                    MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?
1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM
ENTER A VALID NUMBER: 1
You can locate record by TERM, AUTHOR, or NUMBER ([T], A, or N): T
Which TERM do you want to select? PWR
What weight threshold? [>0.]: .5
Use synonyms? (Y or [N]): N
Do you want to return to the MENU now or EXIT the program? (E or [M]): M

                    MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?
1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM
 SET  NO.  QUERY   THRESHOLD
   1   59   PWR        0.5
ENTER A VALID NUMBER:
```

Figure 5. Main Menu with Selection of Option 1.

NEWLINE is pressed without a range, the user returns to the main menu. Fig. 6 shows the use of this option.

Set 2 is a union of the sets retrieved using terms 4 and 5. These terms are so similar in meaning that the union has only 1 more document than the set retrieved by term5, SPECTROM, alone.

With Option 3, the user is asked which logical combination of retrived sets he wants to use. Examples are given to help the user in formulating the query. The choices are to combine sets using AND, OR, or NOT connectives. Of course, this option is available only after two or more sets have been created. Examples of commands using connectives AND, OR, and NOT are: 1*2, 1+2, and 1−2 respectively. It should be noted that NOT is interpreted as the MINUS operator, i.e., the set difference operation.

With this choice, the user has the option to use query weights to convey the importance of terms. For example, the query 1(.2) * 2(1.) shows a lesser importance (.2) for the term(s) associated with set 1 than the importance (1.) of the term(s) associated with set 2. If no query weights are entered, the default weight of 1 is assumed. The program uses the set combination algorithms of Yager as discussed in Sect. 3.2. Fig. 7 shows the use of this option.

With Option 4, the user is asked to construct a query that is a complex Boolean expression. The query contains terms (alphabetic character strings of eight or fewer letters), operators *, +, or − (meaning AND, OR, NOT respectively), and parentheses pairs. Optionally, the expression elements may be separated by one blank. For example,

(term1 * term2)+((term3 * term4)*term5)

is a form of a query accepted by JANE in this option. There must be N−2 parentheses pairs where N is the number of terms; otherwise an error message will result.

A number of sets will be created in the process of responding to the query. The last one is the final retrieved set for the complex query.

With Option 5, the user is asked what set he or she wants to print. Then, in a submenu, the user is asked what limits are wanted. The choices are: (1) no limit, (2) an explicit number of

```
                    MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?
1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM
 SET  NO.  QUERY   THRESHOLD

  1    59   PWR        0.5

ENTER A VALID NUMBER: 2
You can locate record by TERM, AUTHOR, or NUMBER ([T], A, or N): T
Which TERM do you want to select? spect
1 . . . SPECTACU                    . . . 2
2 . . . SPECTER                     . . . 1
3 . . . SPECTRA                     . . . 385
4 . . . SPECTRAL                    . . . 85
5 . . . SPECTROM                    . . . 88
6 . . . SPECTROS                    . . . 27
7 . . . SPECTRUM                    . . . 268
8 . . . SPECULAR                    . . . 2
9 . . . SPECULAT                    . . . 1
10 . . . SPEED                      . . . 13
11 . . . SPEEDS                     . . . 2
12 . . . SPEEDUP                    . . . 1
13 . . . SPEEDUPS                   . . . 1
14 . . . SPENCER                    . . . 3
15 . . . SPENDING                   . . . 1
16 . . . SPENT                      . . . 53
17 . . . SPERICAL                   . . . 1
18 . . . SPHERE                     . . . 32
19 . . . SPHERES                    . . . 17
20 . . . SPHERICA                   . . . 58
Enter desired range of terms (e.g., 1 5) or NEWLINE to return to main menu: 4 5
```

Figure 6. Use of Option 2.

RETRIEVE SPECTRAL (Y or [N]): **y** {entered data may be in lower case}

What weight threshold? ([>0.]) **.5**

Use synonyms? Y or [N] {N was chosen by default}

RETRIEVE SPECTROM (Y or [N]): **y**

What weight threshold? ([>0.]) **.2**

Use synonyms? (Y or [N]): **N**

Do you want to return to the MENU now or EXIT the program? (E or [M]): **M**

MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?

1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM

SET	NO.	QUERY	THRESHOLD
1	59	PWR	0.5
2	89	SPECT	0.2

ENTER A VALID NUMBER:

Figure 6. (continued)

MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?

1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM

SET	NO.	QUERY	THRESHOLD
1	59	PWR	0.5
2	89	SPECT	0.2

ENTER A VALID NUMBER: **3**

Combine sets as follows:
Enter 1*2 to combine sets 1 AND 2.
Enter 1+2 to combine set 1 OR 2.
Enter 1−2 to combine set 1 NOT 2.
Use weights to specify importance of sets, e.g.,
Enter 1(.2)*2(1.)

Enter your command: **1*2**

What weight threshold? [>0.]: **.2**

MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?

1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM

SET	NO.	QUERY	THRESHOLD
1	59	PWR	0.5
2	89	SPECT	0.2
3	3	1*2	0.2

ENTER A VALID NUMBER:

Figure 7. 'Combine Sets' Option.

documents, (3) a few (10% of the set with a maximum of 10), or (4) by the year of publication. If the year of publication is selected, the program prompts for beginning and ending year. This option is shown in Fig. 8.

The user is also asked, with another submenu, to specify the format, i.e., the fields to print. Options are: (1) authors and title only, (2), authors, title, publication description, and publication date, (3) the fields of option 2 plus abstract, (4) the fields of option 3 plus keywords and subject categories (the manual indexing information), and (5) the entire document record. The selected fields are then displayed on the screen and written to the LIST file with the name JANE.OUT. Prior to displaying the results, the set is sorted so that the output will be presented in the order of decreasing retrieval weight. The output file is printed at the user's convenience after exiting from the program. The "Print Format Menu" is shown in Fig. 9.

The first line of each document displayed contains the accession number and the Retrieval Status Value of the document in the set retrieved. Set 3 had 3 documents and all had the same RSV. In this case, only one document contained an abstract. The abstracts are missing because they were not available in English.

5.3 Limitations

The present version allows a total of 15 sets to be retrieved during one session, each consisting of up to 2000 documents. This can easily be revised by changing two parameters in the COMMON file called COMMON_DATA.JANE and recompiling the JANE program (see Programmer's Manual [Tr88]).

MAIN MENU

WHAT TYPE OF OPERATION DO YOU WANT TO DO?

1 RETRIEVE A SET USING TERM, AUTHOR, OR NUMBER
2 RETRIEVE A SET USING A DISPLAY OF KEYS
3 COMBINE SETS
4 RETRIEVE A SET USING A COMPLEX QUERY
5 PRINT RECORDS
6 EXIT THE PROGRAM

SET	NO.	QUERY	THRESHOLD
1	59	PWR	0.5
2	89	SPECT	0.2
3	3	1*2	0.2

ENTER A VALID NUMBER: **5**

Which set? **3**

PRINT LIMITS MENU

WHAT LIMITS ON OUTPUT DO YOU WANT?

1 ... No limit (default).
2 ... Limit number of documents explicitly.
3 ... Limit number of documents to a few.
4 ... Limit documents by date.
5 ... Exit menu.

ENTER A VALID MENU NUMBER. **1**

Document printout is not limited.

Figure 8. Print Records Option.

PRINT FORMAT MENU

WHAT FORMAT DO YOU WANT?

1 ... Authors, Title
2 ... Authors, Title, Publication Desc., Pub. Date
3 ... Above . . . plus Abstract
4 ... Above . . . plus Keywords, Subject Categories
5 ... Complete printout of record

ENTER A VALID NUMBER (A NEWLINE will exit without printing): **3**

11006 Wt: .615
Endres, G.W.R.; Brackenbush, L.W.
Neutron Dosimetry and Spectral Measurements in PWR Containment.
PNL-SA-9461; CONF-810673-4
May 1981
The neutron spectra and absorbed dose rates in reactor containment were measured. These measurements were related to personnel dosimeter response and measurements taken with the neutron monitoring instruments at the nuclear plants. Two different systems were used to measure the neutron spectra. One system, called the multisphere or Bonner shpere system, was composed of 6LiI detector which was placed inside a set of polyethylene spheres of various sizes. The multisphere system is a low resolution system which estimates the spectrum over the full range of neutron energies. The other spectrometer system was composed of a 3He proportional counter filled to 4 psi 3He and 26 psi argon. These 3He counters have a relatively high resolution (2.5 to 3.0%) but cover the more restricted energy range from about 30 keV to > 1 MeV. A pulse shape discrimination system was used to help minimize interference from 3He recoil reactions. A tissue equivalent proportional counter system was used for absorbed dose measurements. Because of the relatively low dose rates at many of the work locations in containment, a 5"-diameter spherical counter was used for most of these measurements. In areas of relatively high dose rate a 2" spherical counter was used.

Figure 9. Print Menu and Output.

12749 Wt: .615
Pineira, T.; Monin, J.P.; Trotabas, M.
Radial Analysis of PWR Fuels by Gamma Spectrometry. Development of Techniques and Application of the TOMOGAM Computation Code.
CEA-CONF-6309 (In French); CONF-8206149-2 (In French)
June 1982

12224 Wt: .615
Pineira, T.; Monin, J.P.; Robert, C.
Radial Distribution of Fission Products in Fuel Pins. Experimental Results on PWR Elements by Gamma Spectrometry. Destructive Testings (ISOPROF Code). Direct Testings (TOMOGAN Code).
CEA-CONF-6005 (In French); CONF-8109138-1 (In French)
September 1981

Figure 9. Print Menu and Output (continued).

6. COMPARISON OF NEW AND PRESENT SYSTEMS

In order to compare the present system and the new JANE system, we will review the features of the ADSEP and ORLOOK programs which have made it necessary to replace these programs with a new system. The review will also facilitate the understanding of the tests made with a number of searches that were requested of RSIC in late 1987.

6.1 ADSEP System

The ADSEP (Automatic Data Set Entry Program) was developed at Oak Ridge for the IBM System 360 prior to 1970. It was adapted at that time for text databases for the many information activities of the Environmental Program of the Oak Ridge National Laboratory. It is a batch program which reads card-image data and stores the data in a hierarchical data structure [Br70]. The program has never been properly documented, but the input requirements are sufficiently simple so that users quickly learn to format their data for input. A sample ADSEP input record is shown in Fig. 3 at the end of Sect. 1. This form is sometimes called the "external ADSEP format."

The usual application of ADSEP is to enter a new record (document) or to replace an entire field in a specified record, although it is possible to create input which will search for a character string in a field and selectively edit the field. The data consists of labels and fields (actual text data). The labels consist of up to eight characters of which the first is an upper case alphabetic character and are enclosed by "<" and ">" symbols. Internal blanks in a label are permitted. Trailing blanks can be included before the ">" symbol, but they are not part of the label. A given label can appear only once in a record.

If a label is numeric, it is taken to be an ADSEP record number which must be in increasing order in the input file. If the the label is null, i.e., "<>", ADSEP will assign the next record number to the record. Multiple blanks in a field are deleted, and line and paragraphing structures are lost. Multiple values in a field, e.g., authors or keywords, are separated by semicolons.

An ADSEP "run", or job, can either add records or revise records (including adding fields), but it cannot do both in a single run.

The main problem in the use of ADSEP is the lack of a facility for interactive editing of the database records. The use of the ADSEP program is phasing out, partly because batch programs for database maintenance are obsolete and partly because there are fewer groups requiring text database service at ORNL.

No attempt was made during the present information retrieval project to develop a substitute for ADSEP, but this will need to be done in the future. Until the entire database is moved to the Eclipse, the database will be maintained on the IBM 3033 mainframe system. After it has been moved, the ADSEP program will no longer be needed, and maintenance will be done entirely on the Eclipse.

6.2 ORLOOK Program

The interactive ORLOOK search program [Si75], an extensive revision of a batch system developed earlier at ORNL, came into use in 1970, and was improved several times in the early 1970s. When RECON became available, further development of ORLOOK was terminated. From the user's point of view, the biggest difference between the two was the use of index files with pointers by RECON. ORLOOK uses a string-matching technique for searching, which is normally confined to particular fields. Not making use of inverted index files makes ORLOOK retrieval very slow. It is limited to a database of 10 000 records (documents).

After the user logs into TSO, the IBM time-sharing control program, he invokes ORLOOK. The program begins by asking the user what file he or she wishes to open and prompting for the IBM system specifications needed to find it, i.e., the dataset name, device (disk type), and volume. At the time this was written, the dataset names were DKT.RSI.VOLX, where X is 1 to 4, the device is 3330-1, and the volume is ZX5555. When ORLOOK gives a "dot" prompt, he or she can enter any of the commands, but will normally enter a "LOOK" request which specifies the fields and strings for a search of set "0", the entire database. If more than one field-string pair are entered on a line, these are taken together as if they are connected by logical AND. Successive lines in a request are interpreted as being combined by a logical OR. The LOOK request is ended with "END."

The result is a set with the next available number; the first search produces set 1. The pointer automatically points to the resulting set, although a command can change the pointer to point to another set, e.g., set 0.

Any set can be searched at any time after it is created. This results in another set, where the first is connected to the second by logical AND. Two sets may also be combined using Boolean logic operators (AND, OR and NOT) using the COMBINE command.

The user can use PRINT or LIST commands to print specified sets on a line printer or the terminal respectively. Particular fields can be specified, including their order. No options are available in format; the line printer uses wide paper and prints in the external ADSEP format. The printout does not have much eye appeal.

The main problems of depending on ORLOOK are (1) slow searches resulting in excessive personnel time and machine costs, (2) the requirement for the database to be stored on the IBM mainframe system which is costly, (3) lack of maintenance and upgrades to ORLOOK and ADSEP, and (4) lack of flexibility in output (no way to change format or sort documents).

6.3 Results of Searches

The RSI data base is stored in four files which were created in successive time periods. The first file consists of the records (6200) which were created and originally maintained at the Oak Ridge Gaseous Diffusion Plant using locally-written software [Pe63, Jo69, Jo71]. The later files were created and maintained by the ADSEP program at the Oak Ridge National Laboratory. The third file, known as Volume 3, was selected as the test file for the JANE system because it is fairly recent, and is more complete, i.e., most of the records have been indexed and contain abstracts. Volume 4 is less complete due to the backlog of abstracts to be entered.

All four of the files have been exported to external ADSEP format and stored on tape, but only Volume 3, consisting of more than 4800 records, has been loaded onto the Eclipse disk. The sequence of programs described in the Programmer's Manual [Tr88] were used to process the data and load the INFOS II data base files for retrieval.

A total of 8 search queries received by RSIC was given to the author to fill the requests. The searches were performed by ORLOOK on all 4 volumes and the results were mailed to the requesters. When the JANE program became operational, these searches were repeated with ORLOOK (so the results could be saved), using only Volume 3, and with JANE. Although ORLOOK can search for strings in any field, the ORLOOK searches were restricted to the subject categories so that the results would reflect the effectiveness of manually assigning subject categories. Searching only the subject category field is the normal practice in using ORLOOK, because it has been assumed that manual "indexing" gives good retrieval results. Furthermore, ORLOOK is very slow in searching through long abstracts.

Although ORLOOK does not make use of index files, these tests are meant to help resolve the issue of whether it is necessary to manually assign the descriptors that are used for retrieval, whether the descriptors (subject categories or keywords) are used to construct an index or not. It will be observed that the ORLOOK queries will be comparatively simple, which is due to the precoordinated nature of the subject categories. This reduces the number of "terms" available and limits the search strategy possibilities, but it brings together automatically the terms needed to define common nuclear engineering concepts. Since the vocabulary is strictly controlled, synonymous "terms" in queries are not needed.

The JANE search strategies have to be quite different. The index terms are derived from words likely to be found in the title or abstract of a document. The searcher must always keep in mind that authors vary greatly in how they discuss a particular subject, and so it is necessary to use synonyms to match the various words that might be used. In addition, it is possible that some of the important terms in defining the context are omitted because the author assumes the reader understands the context and does not think of the possibility that the words may be searched or indexed by a computer.

In the discussions that follow, the search strategies are described for both systems, the results are compared, and the differences are analyzed. A summary of the analysis for each query is given below.

In the search descriptions that follow, please note that an asterisk (*) at the end of a search term indicates an "generic" key. That is, all terms that start with the characters preceding the asterisk will be included in the query. It should also be noted that terms that do not occur in a document have no entry in the INFOS database for that document, that is, the INFOS database is not a complete document-term matrix. Therefore, a threshold weight of 0 in a query will result in retrieval of all documents that have a term weight >0.

- **Query 1: Neutron attenuation for 14-MeV source and concrete shield.**

In the ORLOOK search, a set was created using category '853', which includes all types of concrete. The set contained 38 records (documents). This set was searched using category '111210' (14-Mev neutron source), which resulted in a set of 4 records of which 3 were judged relevant. This set is equivalent to the set ('853' AND '111210').

In the JANE search, the first set created, containing 121 records, resulted from using the term 'concrete' and a threshold weight of 0, i.e., all documents that are indexed by 'concrete.' This was narrowed to 23 records for set 2, using a threshold of 0.5. The third set was created using the term 'D,' based on knowledge that the 14-MeV neutron source is created from a deuterium-tritium (D-T) reaction. With a threshold weight of 0.5, this retrieved 46 documents. Combining either set 1 or set 2 AND set 3 produced 4 records, the same number as the ORLOOK search, but they were not the same documents.

The one document in common was titled "Measurement and Analysis of Leakage Neutron Spectra from SS-316, Concrete, Water, and Polyethylene Slabs with a D-T Neutron Source." In the JANE search, it had the highest weight of the documents retrieved, 0.64. The other three documents retrieved by JANE also had 'D' and 'concrete' in the title. The RSIC analyst indexed them by '111210' but not by '853', hence they were not retrieved by ORLOOK but were retrieved by JANE. This may have occurred because two of the documents retrieved by JANE reported research on streaming in concrete, and the other reported the effect of the oxygen cross section on the penetration of neutrons in concrete. Therefore, the analyst apparently decided not to assign the concrete category. A different analyst might very well have assigned a concrete index.

Two of the documents retrieved by ORLOOK using the subject categories, but not by JANE, had 'deuterium' in the title and thus could have been retrieved by JANE if 'deuterium' had been used as a search term synonym of 'D.' One had 'concrete' in the title, and the other had it in the abstract. The remaining document did not have 'concrete' in the title or abstract, and the only clue in the abstract that the document was about 14-MeV fusion neutrons was that it reported a study of the ELMO Bumpy Torus device. The document could have been retrieved by using one of those words, but the device is one of a kind and one is unlikely to use such specialized terms in a search.

Conclusions: Allowing '14-MeV' as an index term would have made the search strategy much simpler and more effective. The strategy of using 'D' worked except that it also retrieves D-D neutron source documents. The thesaurus should be augmented to include the D-deuterium pair, since failure to use 'deuterium' lost several documents in the JANE search. For this query, we judged some documents on streaming to be relevant, although the manual indexer apparently would not have agreed, or he simply failed to enter the needed category code. The two searches achieved about equal recall (3 vs. 4 relevant documents), but ORLOOK had slightly better precision (1.0 vs. 0.75). Recall and precision are defined in Sect. 3.1.3.

• **Query 2: High energy (60–500 MeV) neutron attenuation in ordinary concrete.**

The subject categories have no way to distinguish the energy of the source neutrons. A range of neutron source energies, however, is included in the category '111220'. The ORLOOK search was performed with the following query: '111220' AND '853100', which requested "spectral neutron sources" combined with "ordinary concrete." A total of 4 documents was retrieved with 3 judged to be relevant. Two discussed research with a 52-MeV proton source which creates neutrons. The third document did not explicitly reveal the energy of the source, but it was identified as a synchrotron, and this would be likely to have neutrons in the energy range of interest. The fourth, irrelevant document was concerned with fusion neutrons.

In the JANE search, the first term that came to mind was 'accelerator', but it was quickly determined that this was in the stop list, probably because it had a large negative TDV. The terms 'MeV' AND 'concrete' were tried, but out of 30 documents retrieved, only 5 were relevant. The

ones retrieved suggested that 'proton,' 'cyclotron,' and 'synchrotron' would be useful. Therefore, the next search used '(proton* OR cyclotron OR synchro*) AND concrete.' This retrieved 7 documents, of which 6 were judged relevant. They included the 3 relevant documents that ORLOOK retrieved and not the non-relevant one. The one non-relevant one retrieved by JANE was concened with reactor neutrons and a proton-recoil detector.

Conclusions: Precision and comparative recall were improved by revising the JANE search strategy. This can be contrasted to the usual ORLOOK search in which the user is unlikely to try to improve the query after analyzing search results. The controlled vocabulary of subject categories is relatively small, making a revised search strategy improbable, and, although titles can be searched at speeds comparable to a subject category search, the search of abstracts is usually not attempted.

The precision of the two results was similar, considering the small sample (0.75 for ORLOOK and 0.85 for JANE), but JANE retrieved more relevant documents (7 vs. 4). This search also demonstrated a need for numeric terms to identify energy regions.

- **Query 3: Gamma-ray spectra in air.**

The subject category vocabulary does not have radiation quantities such as dose or spectra, and so the ORLOOK search employed a query equivalent to 'air AND (data from gamma-ray calculations OR measurements).' Only four documents were retrieved, and since none indicated in the abstract that spectra were reported, they were judged non-relevant. One of these was completely irrelevant in that air was not a material discussed; the document was indexed this way by mistake.

In the first JANE search, a set was created using 'spectrum' and other forms of the word, with a threshold of 0.5. This retrieved 131 documents. An AND combination with 'air' yielded only two documents, so the threshold weight was reduced to 0 for 'spectrum*,' which resulted in a new set with 672 documents. Combining the 672 documents using AND connectives, with 'air' and 'gamma' (using synonyms with gamma), resulted in 14 documents retrieved, with 7 judged relevant, i.e., they reported gamma-ray spectra. In the non-relevant documents, the word "spectra," or a variation of the word, referred to source spectra or neutron spectra, rather than gamma-ray flux density spectra.

The ranking appeared to be effective; there were 6 relevant documents in the first nine presented. The only document that was also in the ORLOOK set was ranked last in the 14 documents retrieved by JANE and was judged not relevant.

Conclusions: This search identified a need for an additional subject category for manual indexing to identify high-energy neutrons. The ORLOOK search did poorly; none of the 4 retrieved documents was judged relevant. On the other hand, the JANE search did comparatively well. It retrieved 7 relevant documents and the precision value was 0.5.

• Query 4: Thermal neutron activation of stainless steels or other structural materials, such as aluminum or titanium.

This was a particularly difficult search for relevant documents. The main problem is that activation detectors are frequently used to measure neutron energy spectra. Another problem is that most documents that treat activation are concerned with fusion devices which is quite a different environment from the thermal neutron reactor environment.

In order to isolate the reactor environment, the ORLOOK search used the subject category '111110,' which specified "fission neutrons" to create the first set. The subject category '534000,' which specifies "activation" has only recently been added to the vocabulary, so the titles and abstracts of the first set were searched for 'activation.' The 25 documents found were then searched for the following categories (OR connection): '861000' (iron alloys and steels), '862000' (nickel alloys), '870022' (titanium), and '870013' (aluminum). This resulted in a set of 5 documents of which 3 were judged relevant. The other two were concerned with activation detectors.

Several strategies were attempted in the JANE searches. The most successful one was essentially the following: 'activation AND (steels OR structural).' This produced a set of 45 documents, of which 8 were judged relevant. The remainder were concerned with fusion or activation detectors. Two of the 8 relevant documents were also found by ORLOOK. The remaining relevant document found by ORLOOK was concerned with activated corrosion products, which is not quite the same as structural material activation, although the corrosion products originate from structural materials.

An attempt to remove the fusion documents using NOT was not very successful, the reason apparently being that the weight of the 'fusion' documents was small, and therefore the resulting retrieval weight of these documents was only slightly reduced. This could be seen in the slight changes in ranking after combining using 'NOT fusion.'

Two additional documents were retrieved for titanium in an additional search using 'titanium AND activation.' These were not retrieved by ORLOOK because they had not yet been indexed, and thus they had no subject categories or abstract entered.

Conclusions: In this search, the ORLOOK precision was better (0.6 vs 0.21), but JANE retrieved more relevant documents (10 vs. 3). With so few retrieved, one would prefer a greater number of relevant documents rather than greater precision.

• **Query 5: Shielding for Am/Be and Cf neutron sources. Particular interest is concrete, especially effect of water content on neutron attenuation.**

In the ORLOOK search, a set was created which was retrieved using the neutron categories equivalent to "spectral sources", "fission sources," or "californium sources" ('111220 OR 111110 OR 870098'). This set contained 433 documents. Combining with the equivalent of "concrete" using AND, resulted in a 10-member set of which 6 were judged relevant. Several of these 6 were borderline; in a fuzzy context, they should be retrieved with a low weight.

A preliminary search with JANE used 'reactor' OR 'fission' OR 'californium,' but it was not very successful, i.e., the precision was very low. The terms 'Am' and 'Be' could not be used because they are in the stop list. This is a real dilemma, because these are important chemical elements, but also very common trivial words. In this case, it was decided that use of these terms was not important (the requester needs generic information on low-energy neutron shielding) and so these terms were not used in the search.

The second JANE search began by locating the 'neutron' documents. This is the most used term in the collection and resulted in a set of 1526 documents. This requirement forced a re-compilation of the JANE program to accomodate a set this large. Combining conjunctively with 'concrete' reduced the set to 54. The terms 'transmission,' 'penetration,' and 'attenuation' were

used disjunctively to create a second set of 246 documents. Noting the similar use of these words, the thesaurus was enlarged at this point to include these terms as being synonyms. Combining the first and second sets, using AND, resulted in a retrieved set of 12 documents, of which 8 were judged relevant. The ranking was effective; 5 of the 8 relevant documents were listed first and two marginal documents were listed among the last three.

Four of the relevant ORLOOK documents were in the JANE result. One of these was marginal and did not mention "concrete," and the other had no word equivalent to "transmission".

Conclusions: This query resulted in similar results from ORLOOK and JANE. The precision was 0.60 and 0.67 respectively, with 6 and 8 relevant documents retrieved.

• **Query 6: For a 2-MeV neutron source, a polyethylene shield is proposed. Data are needed with and without boron added.**

The category '842000' (plastics) was used to begin the ORLOOK search. This category includes polyethylene with or without boron. A reference to 'neutron' is not needed because, for this database, '842000' would normally be assigned to documents that refer to plastic neutron shields. Since only 17 documents were retrieved, these were taken to be the result; the set did not need to be reduced further. No other subject category would add relevant documents. If there had been more documents retrieved, a further search for 'boron' (using AND) would be indicated. There were 6 documents judged to be relevant.

The JANE search again made use of 'neutron,' combining it as follows: 'neutron AND (polyethylene OR plastic*) AND (attenuation AND shield*).' The 'attenuation' term included synonyms. Of the 17 documents retrieved, 14 were judged relevant. They included 3 of the 4 ORLOOK relevant documents; the other relevant document retrieved by ORLOOK mentioned only "hydrogenous" materials, and so it was not picked up by JANE. In performing the combine operation, 'neutron' was given a query weight of 0. With such a low query weight, the documents indexed by 'neutron' are accepted even though their term weights are very small, and the weights of the documents in the retrieved set are determined from weights of other terms. This is accomplished by the Yager algorithm discussed in Sect. 3.2. This procedure results in better performance

in ranking the retrieved documents. All of the first 10 of the retrieved documents were judged relevant.

Conclusions: JANE gave much better results than did ORLOOK for this query. JANE had a precision of 0.82 and retrieved 14 relevant documents, and ORLOOK had a precision of 0.35 and retrieved 6 relevant documents.

• **Query 7: Need to upgrade medical therapy facility. Need neutron shielding data using polyethylene or other plastics.**

The ORLOOK search used a query consisting of the the categories '842000' (plastics) AND '760000' (medical facilities), which retrieved 9 documents. Since there were so few documents retrieved, these were accepted as the result. Only 3 were judged relevant, with 2 having reference to "polyethylene."

The first sets created in the JANE search resulted from use of the terms 'medical' and 'therapy,' with synonyms accepted in each search. To improve recall, the union of these sets was formed, and this set contained 52 documents. Combining this set conjunctively with 'neutron', and giving 'neutron' a very low query weight, resulted in a set of 21.

Of the 21 documents retrieved, 11 were judged relevant if the criteria are relaxed to accept documents treating other shielding materials. Six of the 11 documents were ranked in the first 9; the rest were scattered down the list. About half of these documents were retrieved by terms in the title; the abstract had not yet been entered. Three documents of the 21 retrieved were included in the ORLOOK set, but only one was considered relevant.

Another set of 39 documents was created with the terms 'polyethylene' OR 'plastic*,' which, when combined conjunctively with the 21-document set yielded a set of 2.

Conclusions: Again JANE gave better results than ORLOOK. The precision for JANE was 0.52, and for ORLOOK 0.33. JANE retrieved 11 relevant documents, and ORLOOK 3 relevant documents.

• **Query 8: Shielding analyses of light water reactor plants, especially PWRs.**

The ORLOOK search requested a printout for the subject category '719A00' (light-water reactors), a subject category in a series titled "applications of shield design techniques to complex systems." There were 181 documents indexed by this category. There was no attempt to analyze these documents in regard to being relevant because the query was so broad. It can be assumed that all of the documents are concerned with radiation problems of light water reactors, but not all describe shielding analyses of the plant itself.

The first JANE search produced 147 documents using a search strategy equivalent to 'pressurized' OR 'PWR' OR 'boiling' OR 'BWR.' When this strategy was conceived, it was realized that the equivalent terms should be in the thesaurus, and this was subsequently done. In examining the retrieved documents, it was decided that documents related to the shielding of spent fuel were not really relevant to the query. Therefore, a second search was done which removed most of these documents by combining the set, using NOT, with a set formed from using 'fuel*' OR 'cask*.' This set had 128 documents. It was found that 14 of the 19 removed had been judged not relevant in the first analysis and most of the remaining not-relevant documents were ranked lower, i.e., had a lower retrieval weight.

Although the number of documents retrieved in the ORLOOK and JANE searches were comparable, the overlap was not as great as one might expect. There were only 57 documents that occurred in both the 181-document ORLOOK set and the 147-document JANE set. There are several reasons for this. The decision by the analyst to assign the '719A00' category is not based on well-defined criteria. Some documents had not yet been assigned categories. Finally, the JANE search depended on certain words occurring in the title or abstract.

Conclusions: Both searches retrieved so many documents that a relevance determination was not attempted. Possibly all of the documents can be judged relevant because the query was so broad. ORLOOK retrieved a few more documents than did JANE (181 by ORLOOK and 147 by JANE). About 1/3 of the sets were were common to both.

A summary of the test results is shown in Table 1. The column with the heading 'Comparative Recall' gives the number of relevant documents retrieved.

Table 1. Test Results

Query	Precision		Comparative Recall	
	ORLOOK	JANE	ORLOOK	JANE
1	1.00	0.75	4	3
2	0.75	0.85	4	7
3	0.00	0.50	0	14
4	0.60	0.21	3	10
5	0.60	0.67	6	8
6	0.35	0.82	6	14
7	0.33	0.52	3	11
8	N/A	N/A	181	147

6.4 Conclusions from Systems Comparison

After performing the test searches described above, it appears that the following conclusions can be drawn.

1. **Based on the limited experience so far, it appears that an automatic indexing system can replace the manual indexing used at present by the RSIC staff.** The test cases showed that searching with automatically generated indexing terms can result in precision results comparable to or better than results from searching the manually-assigned subject categories. The recall also appears to be as good, judging from the number of relevant documents retrieved in each search.

2. **The response time for a search on JANE is superior to that of ORLOOK.** The only noticeable wait experienced during the JANE searches was when the set with approximately 1500 documents indexed by 'neutron' was retrieved. On the other hand, search with ORLOOK requires a lot of patience while the entire field specified in the search request is being searched in every document. It should be recalled that ORLOOK cannot make use of index files.

3. **The present simple thesaurus appears to be effective in helping retrieval by expanding the query with synonyms.** This is particularly useful to novice searchers whose technical vocabulary is more limited than those of the professional engineer. It provides a way of building database knowledge into the system. In the searches performed so far, it was found that more

entries should be in the thesaurus. It is expected that the process of thesaurus enhancement will continue as experience accumulates.

4. **The use of a generic search (retrieving all terms beginning with the characters in the query term) helps to find closely related terms such as plural forms or variations in authors' names.**

5. **The concepts associated with fuzzy set theory appear to be useful for ranking documents in the retrieved sets.** It is useful to the user to see the most relevant documents first when a large number of documents is retrieved. The program appears to have had a measure of success in this regard, and it is likely that ranking can be improved with more experience in using query weights.

6. **The chosen algorithms for computing the term weights seem to yield reasonable values.** An occurrence of a term in a title results in a higher weight than one occurrence in an abstract, and this can be observed in the document-matrix. The TDV values markedly reduce the weights of high-frequency terms and increase (somewhat) the weights of mid-frequency terms. The net result is that documents retrieved by mid-frequency terms have a higher rank in the retrieved sets.

7. **The biggest problem with the JANE system at present is the inability to use multiple-term search keys for indexing or retrieval.** It would increase search precision if terms such as "Monte Carlo" or "discrete ordinates" could be used. This problem can be alleviated in a revised system by including word location information to retrieve words that are adjacent, nearby, or in a particular order. This will require more storage, however.

8. **Typographical errors clutter the list of permitted terms.** Errors adversely affect the automatic indexing process and can prevent retrieval of the affected documents. However, if a generic search is used, or if the retrieval option to display terms is used, documents with errors in the latter part of the word can be retrieved. This is also the case with misspelled names. This problem can be largely eliminated if terms with extremely low frequncy in the collection are deleted from the vocabulary.

We conclude from the above that the present version of JANE can replace ORLOOK for searching the RSIC literature database, and offers improvements in the future as experience is gained. The JANE system includes automatic indexing, which offers substantial cost and personnel time savings as compared to the present manual system.

7. CONCLUSIONS

7.1 Summary

It is practical to move the RSIC literature database to the RSIC Data General Eclipse MV/4000 computer and implement a search system based on a locally-written Fortran 77 application program which uses the INFOS II system for database management support. The main issues were (1) practicality of automatic indexing, (2) effective retrieval based on fuzzy-set principles, (3) quick interactive response and (4) sufficient storage capability for the linear file(s) and the INFOS II files. All of these issues appear to be resolved by developing the JANE system and testing it on a large fraction of the total database.

An original algorithm to determine term-document weights and a modified algorithm for the application of Term Discrimination Values were implemented in the JANE automatic indexing process. A published algorithm was used to implement the use of query weights in retrieval, and an original algorithm was adopted to implement set difference in retrieval. An interactive retrieval program was developed and tested using actual queries received by RSIC. The analysis of the test results showed that the new system, JANE, performed as well, and in some cases, better than the currently-used mainframe system based on manual assignment of subject category descriptors.

The present JANE system was developed primarily to resolve the issues listed above. The next stage is to develop a production version of the system. Its main features are discussed in the next subsection.

A final comment concerns portability. The set of programs developed are written in nearly standard Fortran and therefore should be portable to non-Data General systems. The only Data General extensions used were the means to output brackets (non-standard Fortran characters) to the screen and the use of comments on the same line as code. Of course, the INFOS II system is not portable. It would be necessary, on another computer system, to replace the INFOS II functions with another database management system.

7.2. Further Development

For the JANE system to completely replace the ORLOOK system, the entire database will need to be processed and stored on the Eclipse disk structures. For 16 000 documents, it would require 45–50 MB, which could be accomodated without adding to the Eclipse storage capacity. However, this would bring the storage on the present disk uncomfortably close to full use; only about 70 MB of storage now remains on the 350-MB Data General Argus disk. Additional disk acquisition in 1988 is very unlikely, since no capital funds are available to RSIC at present. In the short run, however, a Fujitsu 80-MB disk from an earlier computer could be put into use with the purchase of a $4000 controller. This would not require capital funds.

In order to test the performance of an automatic indexing scheme in comparison with the ORLOOK system, the present JANE system does not make use of the manually-assigned indexing information that is available in the present database. This should undoubtedly be done for a production version so that all information in the file will be used. That is, there should be index records in the INFOS II files for the subject category field and also the logically-equivalent keyword field. Possibly other fields should be indexed as well. For example, access to corporate author or type of publication might be useful.

The system would be improved if term locations in the title and abstract were carried along in the document-term matrix. This would make it possible for adjacency to be required between pairs of terms in a query. The term location information is already available from the program that extracts the terms, but in this initial version of the system, the information is not carried further. The INFOS II database record needs to be expanded to carry this information, and the search program needs to be revised to utilize the information. The INFOS II record would probably need to have varying length to accomodate multiple occurrences of the term in a document. The INFOS II database would be somewhat larger with this information, but this should not be a great problem with care in the design of the INFOS II records. It should be noted that the INFOS II files are already twice the size of the linear file.

It would be desirable to have a more sophisticated method to construct a stem from a word. The present stem is simply the first eight alphabetic characters. Although the use of hyphens

is not consistent, it might be better to include a hyphenated string as a term although that is likely to increase the size of the vocabulary significantly, because all forms would be included, e.g., "gamma," "ray," and "gamma-ray."

It would be desirable to enrich the indexing terms for a document with concepts such as "review article" or with useful classifications such as "shielding formulas." Such terms are not likely to occur in a title or abstract. They would have to be added to the record manually or deduced from some characteristics, e.g., number of terms assigned to a document, or finding mathematical operators.

When it is observed in searching that, to improve recall, the union of sets from equivalent terms is being formed, then these terms should be added to the thesaurus. The thesaurus can be easily revised to accomodate additional synonym pairs. If concepts such as the use of broader and narrower terms are to be implemented, however, the retrieval algorithm logic would need to be revised.

One of the major problems currently is the cost and delay of entering the abstracts manually. It appears that optical character recognition has now reached a point that it is feasible to use that technology instead. The gains in cost and efficiency would be substantial, and an effort will be made in 1988 to make use of this technology.

LIST OF REFERENCES

LIST OF REFERENCES

Ba85 Bärtschi M., 1985, "An Overview of Information Retrieval Subjects," *Computer*, May issue, 67–84.

Bi87a Biswas G., Bezdek J. C., Marques M., and Subramanian V., 1987, "Knowledge-Assisted Document Retrieval: I. The Natural-Language Interface," *Journal of the American Society for Information Science* **38**(2), 83–96.

Bi87b Biswas G., Bezdek J. C., Subramanian V., and Marques M., 1987, "Knowledge-Assisted Document Retrieval: II. The Retrieval Process," *Journal of the American Society for Information Science* **38**(2), 97–110.

Bo80 Bookstein A., 1980, "Fuzzy Requests: An Approach to Weighted Boolean Searches," *Journal of the American Society for Information Science* **31**(3), 240–247.

Bo81 Bookstein A., 1981, "A Comparison of Two Systems of Weighted Boolean Retrieval," *Journal of the American Society for Information Science* **32**(4), 275–279.

Bo85 Bookstein A., 1985, "Probability and Fuzzy-Set Applications to Information Retrieval," pp. 118–151, in: *Annual Review of Information Science and Technology* (edited by M. E. Williams), Vol. 20 (White Plains, N.Y.: Knowledge Industry Publications, Inc.).

Bor82 Borko H., 1982, "Automatic Indexing: A Tutorial." *ACM SIGIR Forum* **16**(2), 9–13.

Boy85 Boyce B. R. and Kraft D. H., 1985, "Principles and Theories in Information Science," pp. 154–178, in: *Annual Review of Information Science and Technology* (edited by M. E. Williams), Vol. 20 (White Plains, N.Y.: Knowledge Industry Publications, Inc.).

Br70 Brooks A. A., 1970, *ADSEP, An Automated Data Set Editing Programing*, CTC-34 (Oak Ridge, Tenn: Union Carbide Corp, Nuclear Division).

Bu85 Buell D. A., 1985, "A Problem in Information Retrieval with Fuzzy Sets," *Journal of the American Society for Information Science* **36**(6), 398–401.

Buc87 Buckles B. P. and Petry F. E., 1987, "Generalized Database and Information Systems," pp. 177–201, in: *Analysis of Fuzzy Information*, (edited by James C. Bezdek), Vol II, Artificial Intelligence and Decision Systems (Boca Raton, Fla.: CRC Press, Inc.).

Ca87 Can F. and Ozkarahan E. A., 1987, "Computation of Term/Document Discrimination Values by Use of the Cover Coefficient Concept," *Journal of the American Society for Information Science* **38**(3), 171–183.

ClC84 Cleverdon C., 1984, "Optimizing Convenient Online Access to Bibliographic Data Bases," *Information Services and Use* **4**(1/2), 37–47.

ClD84 Cleveland D. B., Cleveland A. D. and Wise O. B., 1984, "Less Than Full-Text Indexing Using a Non-Boolean Searching Model," *Journal of the American Society for Information Science* **35**(1), 19–28.

Cr86 Croft W. B., 1987, "Boolean Queries and Term Dependencies in Probabilistic Retrieval Models," *Journal of the American Society for Information Science* **37**(2), 71–77.

Da87 Das-Gupta P., 1987, "Boolean Interpretation of Conjunctions for Document Retrieval," *Journal of the American Society for Information Science* **38**(4), 245–254.

De83 DeJong G., 1983, "Artificial Intelligence Implications for Information Retrieval," in: *Proc. Sixth Annual Int. ACM/SIGIR Conference on Research and Development in Information Retrieval,* ACM SIGIR Forum **17**(4), 10–17.

Di84 Dillon M. and Federhart P., 1984, "Statistical Recognition of Content Terms in General Text," *Journal of the American Society for Information Science* **35**(1), 3–10.

Do86 Doszkocs T. E., 1986, "Natural Language Processing in Information Retrieval," *Journal of the American Society for Information Science* **37**(4), 191–196.

DOE83 *DOE/RECON User's Manual,* 1983, Department of Energy report DOE/TIC-4586-R1 (UPD-2) (Oak Ridge, Tenn.: U.S. Dept. of Energy, Technical Information Center).

Du84 Dutta S. and Sinha P. K., 1984, "Pragmatic Approach to Subject Indexing: A New Concept," *Journal of the American Society for Information Science* **35**(6), 325–331.

Ea85 Eastman C. M., 1985, "Database Management Systems," pp. 103–128, in: *Annual Review of Information Science and Technology* (edited by M. E. Williams), Vol. 20 (White Plains, N.Y.: Knowledge Industry Publications, Inc.).

Ea87 Eastman C. M. and Nakkouzi Z., 1987, "Negation in Queries: Global and Local Contexts," pp. 63–65, in: *ASIS '87, Proceedings of the 50th ASIS Annual Meeting,* (edited by Ching-chih Chen), Vol. 24 (Medford, N.J.: Learned Information, Inc.).

Fi86 Fidel R., 1986, "Towards Expert Systems for the Selection of Search Keys," *Journal of the American Society for Information Science* **37**(1), 37–44.

Gr86 Griffiths A., Luckhurst H. C., and Willet, P., 1987, "Using Interdocument Similarity Information in Document Retrieval Systems," *Journal of the American Society for Information Science* **37**(1), 3–11.

Ha75a Harter S. P., 1975, "A Probabilistic Approach to Automatic Indexing, Part I. On the Distribution of Specialty Words," *Journal of the American Society for Information Science* **26**(4), 197–206.

Ha75b Harter S. P., 1975, "A Probabilistic Approach to Automatic Indexing, Part II. An Algorithm for Probabilistic Indexing," *Journal of the American Society for Information Science* **26**(5), 280–289.

Hu87 Humphrey S. M. and Miller N. E., 1987, "Knowledge-Based Indexing of the Medical Literature: The Indexing Aid Project," *Journal of the American Society for Information Science* **38**(3), 184–196.

IN85 *AOS/VS INFOS II System User's Manual,* 1985, 093-000299-01, ©Data General Corporation (Westborough, Mass.: Data General Corporation).

Jo69 Jones J. G., Trubey D. K., and Gurney J., 1969, *The 1969 RSIC Computer-Based Information Retrieval System in Brief,* ORNL-TM-2719 (Oak Ridge, Tenn.: Oak Ridge National Laboratory).

Jo71 Jones J. G., Green R. O., and Yount S. L., 1971, *Radiation Shielding Information User Documentation,* CTC-48 (Oak Ridge, Tenn.: Union Carbide Corp., Nuclear Division).

Ke62 Kent A., 1962, *Textbook on Mechanized Information Retrieval* (New York: Interscience Publishers, a Division of John Wiley & Sons).

Mi83 Michalski, R. S., 1983, "Learning from Observation: Conceptial Clustering," Chapter 11, pp 331–363, in: *Machine Learning: Artificial Intelligence Approach* (edited by R. S. Michalski, J. G. Carbonell, and T. M. Michell), (Palo Alto, Calif.: Tioga Publishing).

Pe63 Penny S. K., Trubey D. K., and Emmett M. B., 1963, "Radiation Shielding Information Center Information Retrieval System," pp. 251–252, in: *ADI Annual Meeting 1963, Short Papers,* Part 2 (Washington, D. C.: American Documentation Institute).

Pr87 Prade H. and Testemale C., 1987, "Fuzzy Relational Databases: Representational Issues and Reduction Using Similarity Measures," *Journal of the American Society for Information Science* **38**(2), 118–126.

Ra82 Radecki T., 1982, "A Probabilistic Approach to Information Retrieval in Systems with Boolean Search Request Formulations," *Journal of the American Society for Information Science* **33**(6), 365–370.

Ra85 Radecki T., 1985, "Document Ranking Methodology for Boolean Retrieval Systems," pp. 265–272, in: *Empirical Foundations of Information and Software Science, Proceedings of the Second Symposium on Empirical Foundations of Information and Software Science,* Georgia Institute of Technology, held Oct. 3–5, 1984, Atlanta, Ga., edited by Jagdish C. Agrawal and Pranas Zunde (New York, N.Y.: Plenum Press).

Ra86 Raghavan V. V. and Wong S. K. M., 1987, "A Critical Analysis of Vector Space Model for Information Retrieval," *Journal of the American Society for Information Science* **37**(5), 279–287.

Sa75 Salton G., 1975, "A Theory of Indexing," in: *Regional Conference Series in Applied Mathematics* No. 18 (Philadelphia, Pa.: Society for Industrial and Applied Mathematics).

Sa81 Salton G., Wu H. and Yu C. T., 1981, "The Measurement of Term Importance in Automatic Indexing," *Journal of the American Society for Information Science* **32**(3), 175–186.

Sa83 Salton G., and McGill M. J., 1983, *Introduction to Modern Information Retrieval* (New York: McGraw-Hill).

Sa86 Salton G., 1986, "On The Use of Knowledge-Based Processing in Automatic Text Retrieval," pp. 277–287, in: *ASIS '86, Proceedings of the 49th ASIS Annual Meeting,* (edited by Julia M. Hurd), Vol. 23 (Medford, N.J.: Learned Information, Inc.).

Se83 Sedgewick R, 1983, p. 110, *Algorithms* (Reading, Mass.: Addison-Wesley Publishing Co.).

Sc86 Schwartz C. and Eisenmann L. M., 1986, "Subject Analysis," pp. 37–61, in: *Annual Review of Information Science and Technology* (edited by M. E. Williams), Vol. 21 (White Plains, N.Y.: Knowledge Industry Publications, Inc.).

Sh86 Shepherd M. and Phillips W. J., 1986, "The Profile-Query Relationship," *Journal of the American Society for Information Science* **37**(3), 146–152.

Si75 Singletary V. A., 1975, *An Online Conversational Retrieval System for ORCHIS Text-Oriented Data Bases, User's Manual,* CTC-34 (Oak Ridge, Tenn.: Oak Ridge National Laboratory).

Si83 Sievert M. and Boyce B. R., 1983, "Hedge Trimming and the Resurrection of the Controlled Vocabulary in Online Searching," *Online Review* **7**(6), 489–494.

SO85 *Sort/Merge with Report Writer, User's Manual,* 1985, 093-000155-02, ©Data General Corporation (Westborough, Mass.: Data General Corporation).

Th86 Thornburg G., 1986, *Where Expert Systems Intersect With Information Retrieval: Issues and Applications,* IS 86–15, Reports of the Intelligent Systems Group, Dept. of Computer Science (Urbana, Ill.: University of Illinois at Urbana-Champaign).

Tr67 Trubey D. K., 1967, "The Radiation Shielding Information Center," *Nuclear News* **10**, 24–25.

Tr88 Trubey D. K., 1988, *The JANE Information Retrieval System: Programmer and User Manuals,* To be published (Oak Ridge, Tenn.: Oak Ridge National Laboratory).

Vi87 Vickery B., 1987, "Information Retrieval," pp. 613–625, in: *Encyclopedia of Physical Science and Technology,* Vol. 6 (New York: Academic Press).

Wi85 Willett P., 1985, "An Algorithm for the Calculation of Exact Term Discrimination Values," *Information Processing and Management* **21**(3), 225–232.

Ya87 Yager R. R., 1987, "A Note on Weighted Queries in Information Retrieval Systems," *Journal of the American Society for Information Science* **38**(1), 23–24.

Za65 Zadeh L. A., 1965, "Fuzzy Sets," *Information and Control* **8**, 338–353.

APPENDICES

APPENDIX A

JANE Program Structure

The JANE retrieval program consists of a number of subroutines to carry out various functions which are briefly described below and in more detail in the ORNL report on the system [Tr88]. The subroutine structure is shown in Fig. 10.

• MAIN

The main routine opens the INFOS II index file and checks to see if the INFOS II file structure is built; if not, it writes the three selector keys and defines a subindex under each. The selector keys, TERMS, AUTHORS, and NUMBERS, and the subindexes for each are shown in Fig. 4. This must be done before the database records are loaded. It then loads the file THESAURUS into an array. Each term in the thesaurus may have up to 3 synonyms. In the present version, up to 100 terms (dimensioned in the subroutine THESAURUS) can be loaded. It then calls MENU to allow the user to select an operation, and then calls the subroutine needed to carry out the function.

Subroutines Called by MAIN (alphabetical order):

• COMPLEX

Called from the Complex Query option, it asks for the command and calls PARSE to analyze the command and GET_SETS to retrieve and combine the sets as needed.
Supporting routine called: UPPER.

• COMBINE

This routine asks the user to input a combine command in one of the following forms:

1*2 meaning combine sets 1 AND 2,

1+2 meaning combine sets 1 OR 2,

1−2 meaning combine sets 1 MINUS 2.

Query weights may be included. It checks for input errors and, if none, calls one of the following routines.

COM_AND This routine performs intersection on the 2 sets.

COM_OR This routine performs union on the 2 sets.

COM_NOT This routine performs set difference on the 2 sets.

Supporting routines called: GET_NUM, GET_REAL, UPPER.

• DUPLICATE

This routine eliminates duplicate pointers to a document and retains the larger weight of the document-term pair.

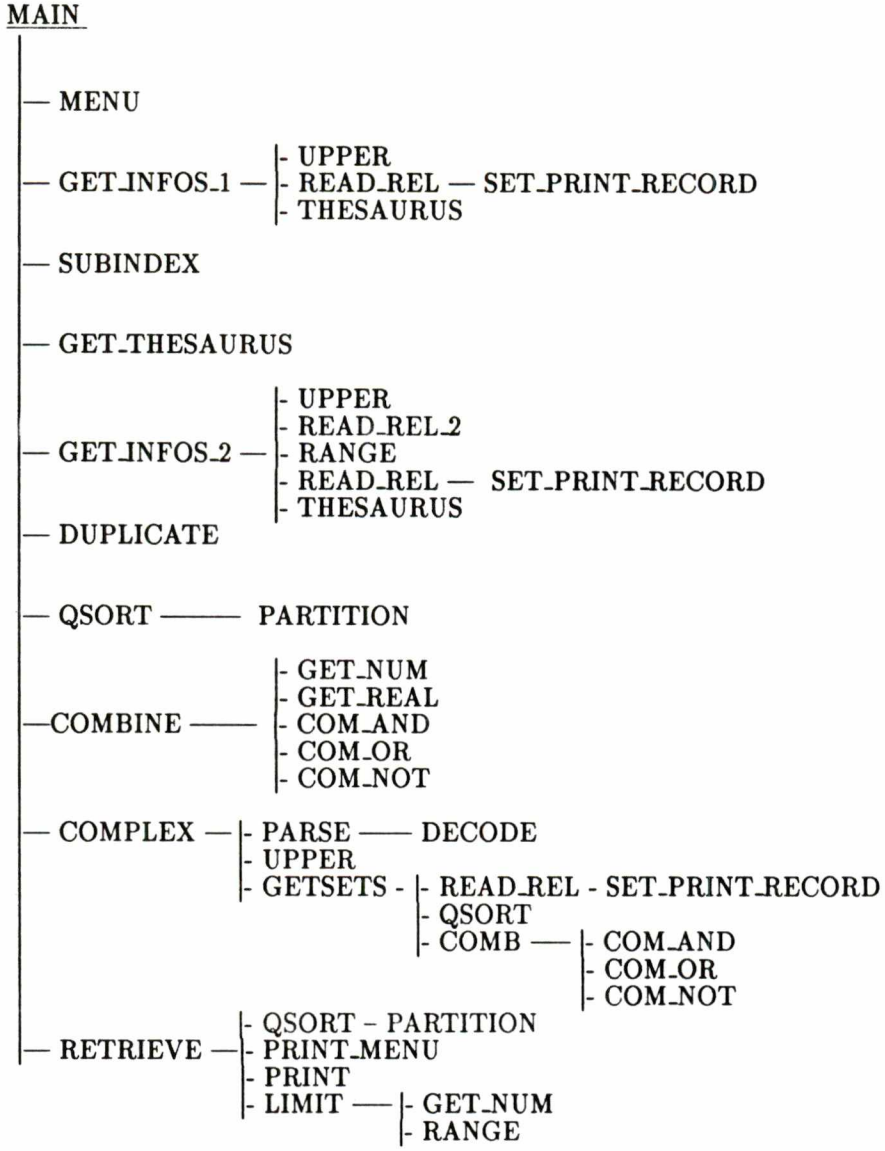


Figure 10. JANE Retrieval Program Subroutine Structure.

• **GET_INFOS_1**

Called from the main routine in either of the first 2 options, it asks the user for a term and weight threshold (default: >0), author, or accession number. It then calls READ_REL to access the INFOS II files and retrieve the desired records. The user can optionally add terms from the thesaurus. This is effected by calling THESAURUS.
Supporting routines called: READ_REL, UPPER.

• **GET_INFOS_2**

This routine, called from the main routine with the 'display' option, displays search terms, authors, or accession numbers for the user to create a query using OR connectives of the displayed keys. The user can select a range of keys and then select individually from the range. The user can scroll down through list.
Supporting routines called: READ_REL_2, READ_REL, RANGE, THESAURUS, UPPER.

• **GET_THESAURUS**

The thesaurus is read by this routine and stored as arrays of terms and pointers to the beginning of each initial letter in the term array. The purpose is to aid the lookup of terms by the subroutine THESAURUS.

• **MENU**

This routine displays the primary options to the user. In addition to the options, data on the existing retrieved sets are displayed.

• **QSORT**

This routine sorts an array of integers or real numbers using the an iterative (non-recursive) QuickSort algorithm given by Sedgewick [Se83]. It calls a function, PARTITION, to partition the array in the sort process.

• **SUBINDEX**

This routine asks the user whether he/she wants to search by term, authors, or number.

• **RETRIEVE**

The routine asks the user which set to print and then opens the direct access file RSI.RAN to extract the desired fields from the file. The set is first sorted by retrieval weight using QSORT and calls LIMIT to ask the user if he/she wants to limit the output. PRINT_MENU is called next to determine what fields to print. The stored set has pointers to the header record of each document in the direct-access file, and each header record has pointers to the most-accessed fields. This routine makes use of the pointers to the following fields: AUTHOR, TITLE, PUB DESC, PUB DATE, ABSTRACT, and SRCHDATE. Other available pointers are for the fields KEYWORDS and SUB CAT. The retrieved document fields are displayed on the screen and written to the LIST file for later printing if desired.
Supporting routines called: QSORT, LIMIT, PRINT_MENU, PRINT.

Supporting Subroutines:

- **COMB**

This routine calls the proper routine to combine the sets.
Called by GET_SETS. Routines called: COM_AND, COM_OR, COM_NOT.

- **DECODE**

This routine is called by PARSE to decode the characters in the command string and return the various elements, i.e., terms, operators, and parentheses.
Called by PARSE.

- **GET_NUM**

This routine extracts an integer from a string ended by a non-numeric character.
Called by COMBINE, LIMIT.

- **GET_REAL**

This routine extracts a real number from a string ended by a non-numeric character.
Called by COMBINE.

- **GET_SETS**

This routine calls READ_REL to obtain a set for each term and COMB to combine the sets as needed.
Called by COMPLEX.
Routines called: READ_REL, QSORT, COMB.

- **LIMIT**

A menu is displayed, giving the user the following options: (1) no limit (default), (2) limit specified specifed by user, e.g., 100, (3) limit to a few (the maximum of 10% of the set, or 10), (4) limited by publication date (year). Option 4 can be combined with the earlier options.
Called by RETRIEVE.
Routines called: GET_NUM, RANGE.

- **PARSE**

This routine separates terms and operators from input and places them in array structures for later processing. The query consists of the operators $*$, $+$, and $-$, terms, and parentheses pairs. One blank may separate elements.
Called by COMPLEX.
Routine called: DECODE.

- **PRINT**

This routine is called for each document to be printed. It accesses the linear file and displays

the selected fields on the screen and writes them to the output file.
Called by RETRIEVE.

- **PRINT_MENU**

The menu is displayed, giving the user the following format options: (1) authors, title, (2) authors, title, publication description, publication date, (3) the previous plus keywords and RSIC subject categories, (4) the entire record (document).

- **RANGE**

Two integers are extracted by this routine from a string. If only one is present, the second integer is taken to be the same as the first.
Called by GET_INFOS_2, LIMIT.

- **READ_REL**

This routine retrieves a set using a key. It stores the direct access numbers in the 2-dimensional array SETS. The following INFOS II interface routines are called: IKEYREAD to use keyed access to the INFOS II record and IRELREAD to use relative access to traverse the keys while they satisfy the query. Other routine called: SET_PRINT_RECORD.
Called by GET_INFOS_1, GET_INFOS_2, GET_SETS.

- **READ_REL_2**

This routine uses a keyed read of the INFOS II index and uses relative motion to traverse the keys and retrieve them. It passes a list of keys, normally 20, and their postings to GET_INFOS_2.
Called by GET_INFOS_2.

- **SET_PRINT_RECORD**

This routine is called by READ_REL for each retrieved record. It accepts the item for the retrieved set if the weight exceeds the threshold, and displays the retrieved record on the screen.
Called by READ_REL.

- **THESAURUS**

This routine will return a synonym from the thesaurus array. Each term may have up to 3 synonyms. The routine requires a synonym number (1–3) as a parameter to specify which synonym is to be returned. The subroutine returns a blank string if there is no synonym in the position specified. For fast look-up, pointers are used to locate the section of the thesaurus array with the same initial letter. A binary search then locates the term if it is there.
Called by GET_INFOS_1, GET_INFOS_2.

- **UPPER**

This routine converts any lower case letters in the COMMON variable TEMPSTRING to upper case. All search keys are upper case, and this routine is used to convert lower case input.

APPENDIX B

Programs for Creation of INFOS II and Linear Files

The following programs are run in sequence to process the database for use in the JANE system.

BREAK

The BREAK program operates on the linear file received from the IBM mainframe to assure that words are not broken at column 80.

RANDOM

The RANDOM program reads a sequential external ADSEP formatted file and writes a direct (random) access file with the same data but with each record having a length of 80 bytes.

EXTRACT

The EXTRACT program reads a direct access linear file and extracts the information needed for retrieval. The output is a file of records giving document-term pairs with the document accession number and direct access number.

ALENGTH

The ALENGTH program reads a file output from EXTRACT and adds to each record the length of the abstract, if the word was extracted from an abstract.

SORT/MERGE (1)

The Data General SORT/MERGE utility is invoked to sort the EXTRACT output by term and accession number.

FREQ

The FREQ program reads the EXTRACT file and the STOP.LIST file, computes the term weights, and outputs the records for the INFOS database.

SORT/MERGE (2)

The Data General SORT/MERGE utility is invoked to sort the FREQ output by accession number and term.

ALPHA

Several parameters are computed to implement the Cover Coefficient method of Can and Ozkarahan which is used to compute the term discrimination value (TDV) for each term. This program computes α.

BETA

The next parameter, to be computed to implement the Cover Coefficient method of Can and Ozkarahan is β. This program computes β.

DELTA

The final parameter to be computed before the term discrimination values are computed is δ. This program computes δ.

TDV

The Term Discrimination Value (TDV) for each term is computed according to the final part of the Can and Ozkarahan algorithm.

SORT/MERGE (3)

The document-term data are inserted into the INFOS database by SORT/MERGE.

EXNAME

The EXNAME program extracts authors' names from the direct access linear file and creates records for the INFOS database.

SORT/MERGE (4)

The author data are inserted into the INFOS database by SORT/MERGE.

The final step is to create the number keys from the records in the INFOS II database. SORT/MERGE is again invoked to accomplish this. This completes the creation of the data files.

VITA

David Keith Trubey was born in Coldwater, Michigan on April 23, 1928. He graduated from Colon (Michigan) High School in 1946 and attended Lawrence Institute of Techology in Detroit for two years. After a tour of duty in the U.S. Navy as a naval aviator flying P2V Neptune patrol bomber aircraft during the Korean war, he attended Michigan State University from which he graduated in 1953 with a Bachelor of Science degree in Physics. In 1983, he re-entered the Graduate School of the University of Tennessee, Knoxville, and received the Master's degree in Computer Science in August, 1988.

In September 1953, he joined the research staff of Oak Ridge National Laboratory (ORNL) and performed radiation shielding experiments before attending the Oak Ridge School of Reactor Technology (ORSORT). During the 1950s, Trubey performed radiation transport and shielding calculations using the ORACLE (Oak Ridge Automatic Computer and Logical Engine) and taught the shielding course for several years at ORSORT in the early 1960s.

Trubey was one of three founding staff members of the Radiation Shielding Information Center (RSIC) in 1962 and became manager in 1966. He left RSIC for one year in 1970 to found an information activity in the ORNL Environmental Program. After his return to RSIC, he guided the ever increasing use of computers by RSIC staff for information processing and publishing. He supervised the acquisition of a Data General Eclipse S/130 computer in 1979 and a Data General Eclipse MV/4000 machine in 1985, and has acted as system manager for these machines.

Since 1953, Trubey has been author or co-author of more than 150 papers, articles, technical reports, and book chapters on radiation transport and shielding topics. He served for several years on the editorial board of *Nuclear Engineering and Design* and serves as reviewer for *Nuclear Science and Engineering*. He is a Fellow of the American Nuclear Society (ANS) and member of the Health Physics Society, Sigma Pi Sigma, Phi Kappa Phi, Sigma Xi, and Upsilon Pi Epsilon. In 1977, He organized and was Chairman of the Fifth International Conference on Radiation Shielding, the only conference in this series to be held in the United States.

Trubey was Chairman of the ANS Radiation Protection and Shielding Division in 1974–75 and has held other offices in the Division. He has been active in standards development in the ANS and is Chairman of the ANS Standards Subcommittee on Radiation Protection and Shielding and member of ANS consensus committee N17, Research Reactors, Reactor Physics, and Radiation Shielding.

In 1950, he married the former Jean Marie Mains. They have two sons, Robert, an artist, and Richard, a research chemist.