FG

12

RADC-TR-76-226
Final Technical Report
July 1976

APPLICATION OF PATTERN ANALYSIS AND RECOGNITION TO I & W

Pattern Analysis and Recognition Corp.

Approved for public release;
distribution unlimited.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED:

SAMUEL S. DICARLO
Project Engineer

APPROVED:

HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
~~1. REPORT NUMBER~~ (18) RADC-TR-76-226	2. GOVT ACCESSION NO.	~~3. RECIPIENT'S CATALOG NUMBER~~ (9)
4. TITLE (and Subtitle) (6) APPLICATION OF PATTERN ANALYSIS AND RECOGNITION TO I & W. 7 and 7		~~5. TYPE OF REPORT & PERIOD COVERED~~ Final Technical Report. ~~March 1975 - January 1976~~
		~~6. PERFORMING ORG. REPORT NUMBER~~ PAR Report No. 76-3
7. AUTHOR(s) (10) [redacted] Christopher Landauer, ~~Dr.~~ John Sanders [redacted] John Morris, ~~Mr.~~ Frank Blackburn [redacted] Clinton Mah, Mr. David Bennett		~~8. CONTRACT OR GRANT NUMBER(s)~~ F30602-75-C-0157 NEW (15)
~~9. PERFORMING ORGANIZATION~~ NAME AND ADDRESS Pattern Analysis and Recognition Corp 228 W Dominick St Rome NY 13440		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45941226
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDA) Griffiss Air Force Base NY 13441		(11) ~~12. REPORT DATE~~ July 1976 (12) 68p. 13. NUMBER OF PAGES 74
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) ~~Same~~ (14) PAR-76-3		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved ~~for public release; distribution~~ unlimited. (16) AF-4594 (17) 459412		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Samuel S. DiCarlo (IRDA)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Information Retrieval Associative Indexing Management Information Systems		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The goal of the initial three-month feasibility study reported here has been the design and testing of critical components of the Message Extraction Through Estimated Relevance (METER) System. Description of the designs and tests are included in this report. In addition, background materials concerning the goals of the project are		

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390101 1B

presented, together with a description of associative methods for document analysis and classification. Finally, a theoretical discussion introduces questions for which additional research will be required.

Reference is made to the RADC Automatic Document Classification On-Line (RADCOL) system, which furnished initial designs for METER components. An evaluation of the RADCOL system is provided in RADC-TR-75-208.

RADC-TR-76-226

EVALUATION

The central issue addressed in this effort was the feasibility of designing an efficient system for processing a large volume of messages under severe time constraints on a minimal equipment configuration.

This effort is of particular value in that it indicates that a correlative approach can be more efficient than clustering algorithms and standard factor analysis techniques that are unyielding or inapplicable when 4,000 to 5,000 word stems are used. Second, the RADCOL System clustering algorithms depended heavily on chance and clusters were frequently found to to be badly formed. Third, the Direct Linkage Algorithm, as conceived during the course of this study, offers substantial savings in memory requirements. Finally, significant savings in "set-up" time is possible making this approach practical for processing messages and providing an associative retrieval capability as well as key-word and Boolean retrievals.

SAMUEL S. DICARLO
Project Engineer

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

The goal of the initial three-month feasibility study reported here has been the design and testing of critical components of the Message Extraction Through Estimated Relevance (METER) System. Descriptions of the designs and tests are included in this report.

In addition, background materials concerning the goals of the project are presented, together with a description of associative methods for document analysis and classification. Finally, a theoretical discussion introduces questions for which additional research will be required.

1.1. BACKGROUND

Methods for document analysis based on statistical characteristics of word-distributions have been under study for several years. RADC-sponsored projects have included the Classification Space Analysis of P. G. Ossorio reported in RADC-TDR-64-287, October 1964; and studies of the Mathematics of Information Storage and Retrieval by J. W. Sammon, described in PAR TR-69-23, June 1969. The most extensive and significant of these projects has been implementation of the RADCOL (RADC Automatic Document Classification On-Line) system, which is briefly described and evaluated in RADC-TR-75-208, August 1975. PAR has also applied the On-Line Pattern Analysis and Recognition System (OLPARS) to the classification of Indications and Warnings (I&W) messages in a series of experiments reported in RADC-TR-75-70.

The goal of the METER development is somewhat different from those of the earlier projects, in that it aims at the design of a system for the processing of a large volume of messages within the constraints of current intelligence analysis. The METER project also contrasts sharply with earlier work which utilized considerably larger computer equipment operating under less severe time constraints. In addition, it contrasts with academic developments, which have used much smaller, static data bases.

Associative message analysis serves to expand the capabilities of traditional retrieval systems, by providing the ability to search for documents within a subject-matter area, which has been structured statistically on the basis of word associations within the data. Because associative analysis differs in concept from traditional information retrieval, it is described in detail in Section 2 of this report.

1.2. PARAMETERS OF THE PROBLEM

Current intelligence is faced with the problem of responding promptly and appropriately to an increasing flood of information. In the initial feasibility studies described here, it was assumed that an average of 5000 messages per day would require processing, with a maximum of 7000 messages per day. Since the maximum figure would be likely to occur during a crisis, when message processing would be most urgently needed, it was essential to plan for adequate processing facilities at the maximum figure, rather than the average.

A five-day file of messages would be maintained, with storage provided for a maximum of 35,000 messages. Essential statistics would be recomputed on a daily basis. Again, it would be essêntial to maintain adequate performance at the maximum rate.

In all cases, system design should provide for easy modification of these parameters to permit more frequent updates, larger message files, and storage over a longer period.

It was initially decided that the implementation would use a configuration of medium-sized computers, such as the DEC PDP 11/45. It is expected that final system design will call for two 11/45 or 11/70 computers, of which one will be used for initial message processing and daily updates, while the other will provide interactive processing for five or more analyst stations. The second processor could be running almost 24 hours a day.

The choice of a medium-sized computer for system implementation has required a careful study of computational techniques which would permit extensive statistical processing of the data within a reasonable time. One of the problems, for example, was the development of an efficient integer square root algorithm. Efficiency was essential, since it might be invoked as many as 12 thousand times during system startup. The proposed algorithm is described in Section 4.5 of this report.

The advantages of a successful medium-sized computer implementation lie in providing an extremely sophisticated statistical processing technique on equipment which is widely available at reasonable cost.

Specific design studies completed during the initial feasibility study include the following:

o Fixed point square root algorithm.

o Improved stemming (de-suffixing) methods.

o Design of statistical filter (to select most useful words for discriminating between messages) for volatile data bases.

o Comparison of correlation coefficients.

o Implementation of test data bases.

- Planned implementation of the algorithm design language, PASCAL.

In addition, a variety of programs for word counts and other statistical information derived from the data bases, tests of computational methods and timings, service programs, and other studies have been completed.

1.3. SUMMARY OF THE KEY FEATURES OF THE METER SYSTEM

1.3.1. Message Acceptor: Realtime Initial Processing System and Message Analyzer: Daily Associative Message Analysis System

The key elements of the METER system which will allow start-up to take an estimated 8 to 10 hours (instead of the more than 80 hours required for the original RADCOL system) are as follows:

- The elimination of the clustering routine.

- The performance of stemming on each message as it comes in.

- The calculation of a partial Dennis measure which allows the statistical content measure (Dennis measure) of a stem over the 5-day data base to be computed as a sum of partial measures for each day.

- The computation of correlation in a piecewise manner. This allows the correlation of two stems over the five-day data base to be computed (in most cases) as a sum of partial correlations for each day.

These features will be described in more detail in Section 3.

1.3.2. Message Search: Statistical On-Line Message Extraction System

The unique feature which will allow search times to be reduced to a few seconds is as follows:

In the original RADCOL system the measure of closeness between a document and a query was calculated by first expanding the stem-frequency vector of the query to create a "concept vector" for the query. This calculation was performed by computing Cq, where C is the concept/content stem correlation matrix and q is the content stem-frequency vector for the query, as a column vector. The same calculation was done for each document; i.e., the document concept vectors Cd were calculated during start-up for each document, where d is the stem-frequency vector for a document. (This is another start-up procedure which is avoided in the current design.) The measure was then computed by calculating the inner product $(Cq) \cdot (Cd)$.

The saving in the proposed system comes from noting that since $u \cdot v = u^t v$ for column vectors u,v, we have $(Cq) \cdot (Cd) = q^t(C^t C)d$.

Since we treat every content stem as a cluster center, the C matrix in the METER system is a (square) symmetric matrix so that $C^tC = C^2$. In the proposed system we calculate $q^tC^2 = (C^2q)^t$ at retrieval time and then run through the document-frequency vectors d performing the inner product calculation $(C^2q)\cdot d$. This is a much more efficient way of calculating the metric because the number of non-zero elements in d is much less than the number of non-zero elements in Cd. Note also that the storage space required for storing the d vectors is much less than the space needed to store the concept vectors Cd, as was required in the RADCOL system.

1.4. PROGRAM PLAN

The overall schedule for the design and implementation of a prototype message analysis system is as follows:

I. Design Phase (Sept 75 - June 76)

A. Detailed determination of requirements:

1. Files required

2. Tables

3. Display routines

4. Time-sharing requirements (need for re-entrant routines, etc.)

5. Mass storage requirements

6. Memory requirements (estimated at 64K words)

7. Additional hardware requirements

8. Detailed program specifications

B. Data Base Implementation

Obtain, implement, and review required data bases:

1. Secure simulated messages for system tests

2. Convert to machine-readable form

3. Test against coding requirements for available equipment

4. Obtain statistical data concerning message lengths, word counts, word frequency distributions, etc.

5. Prepare sample queries and develop designs for

interactive experiments with the system

II. Implementation Phase (July 76 - Dec 76)

1. Obtain hardware as required

2. Implement and test Acceptor (initial message preprocessor)

   a. Header/text split and store routines
   b. Stemming and micro-concordance (stems in each document)

3. Implement Analyzer (statistical analysis and data base structuring)

   a. Statistical filter (Dennis measure) on stems

   b. Identify content stems

   c. Stem-stem correlations

   d. Document vector formation

   e. Storage of messages

4. Implement on-line system

5. Develop further experimental designs

III. Experiment Phase (Jan 77 - June 77)

1. Determine most appropriate values for cut-off points for correlations, length of document vectors, disk storage parameters, etc. (i.e., tradeoffs for optimum time/accuracy of retrievals)

2. Determine times required for system startup and update. Estimate times for analysis/retrieval

3. Reprogram as necessary to improve timings

IV. Redesign Phase (July 77 - Dec 77)

1. Determine weaknesses and inefficient operations in prototype implementation

2. Redesign program logic as required

V. Production System Design Phase (Jan 78 - June 78)

VI. Production System Test Phase (July 78 - Sept 78)

SECTION 2

BASIC METHODS

Associative message analysis provides an approach to the solution of several problems involving natural-language documents or messages:

1. Automatic identification of messages dealing with a specific subject-matter area.

2. Rating or ordering of messages according to their relevance to an analyst's interests.

3. Identification of messages which are similar in content to a selected message.

4. Determination of changes or trends in message contents over time.

5. Classification of messages into categories which are determined empirically, on the basis of message contents.

Because of its applicability to problems like these, METER is not referred to here as an "information retrieval" system; its unique capabilities extend beyond those ordinarily available through a retrieval system. The METER system is expected nevertheless to provide all the capabilities of the "standard" retrieval systems now available.

2.1. KEYWORD SYSTEMS

Traditional retrieval systems will be termed "Boolean" or "keyword" systems in this report. This reflects their "all or nothing" character--either a document belongs to the set of retrieved documents or it does not. It also reflects the ability, found in most traditional systems, to use Boolean operators such as AND, OR, and NOT to combine the words that form the queries.

Associative analysis differs from keyword retrieval in the following way. Typically, a keyword retrieval system (Figure 2-1) searches for a particular word in the data base. When an exact match for the keyword is found in a document, then that document is retrieved. Retrieval logic is extremely simple.

Most keyword systems permit the user to expand the search in various ways. Instead of searching for exact matches, the user may be permitted to search for all words that begin with the same few letters. In one such system, by entering COMPUT*, the user is able to retrieve documents containing the words COMPUTER, COMPUTED, COMPUTATION, and so

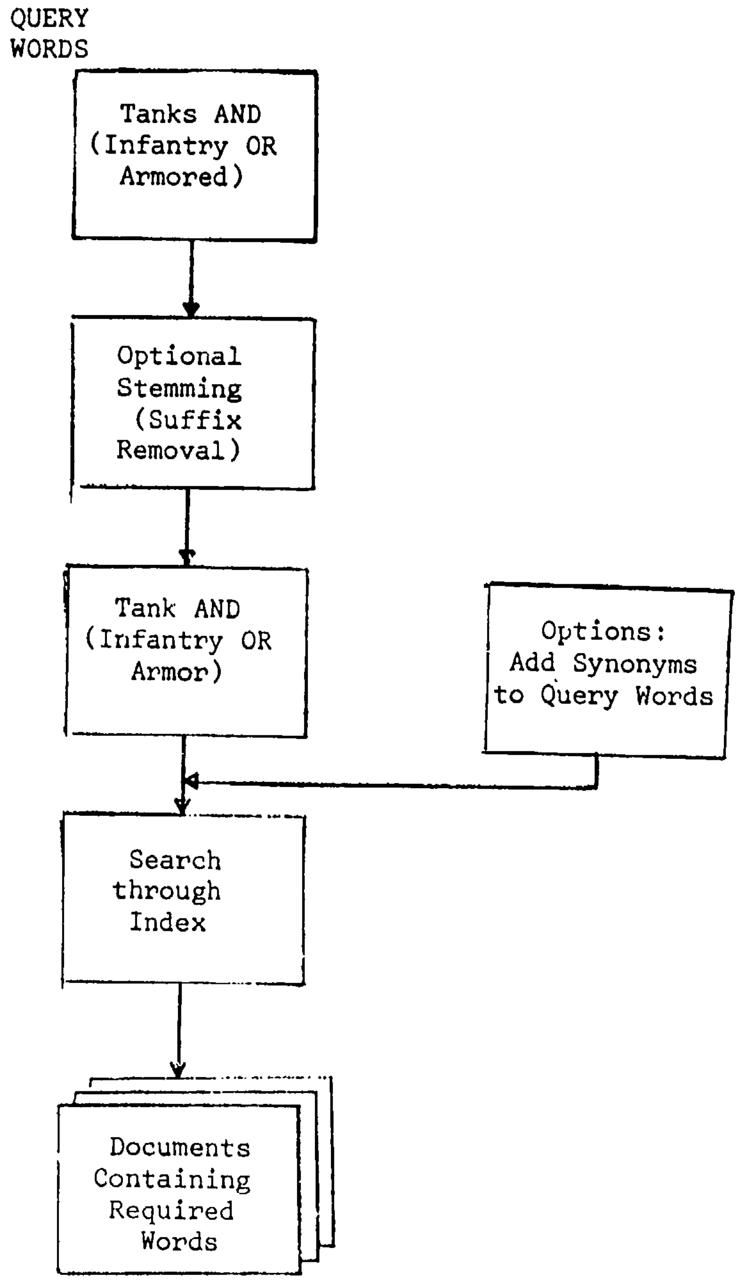

QUERY
WORDS
Tanks AND (Infantry OR Armored)
Optional Stemming (Suffix Removal)
Tank AND (Infantry OR Armor)
Options: Add Synonyms to Query Words
Search through Index
Documents Containing Required Words

Figure 2-1 Boolean Retrieval

on. Various words can be combined together, using AND, OR, and NOT, to obtain documents with the required (Boolean) combination of words. Finally, some systems permit the automatic use of a thesaurus, which will expand a search to include synonyms; in addition to documents which contain the word COMPUTER, these systems may also retrieve documents containing the words DATA PROCESSING.

Boolean or keyword retrieval systems, like those described above, are regarded as standard in the industry.

2.2. ASSOCIATIVE MESSAGE ANALYSIS

Associative analysis (Figure 2-2) does not attempt to retrieve documents which contain particular words. Instead, it searches for documents on the basis of statistical associations among words. When two words tend to appear together in the same documents, they are said to be associated or correlated with one another. If the word COMPUTER frequently appears with the word DATA, then these words will tend to have a high correlation. Later, when the user requests documents containing the word COMPUTER, the system will assume that documents containing the word DATA are also wanted. These correlations permit the system to rank the documents in terms of their potential relevance to a query. Documents which contain words which are highly correlated to the words in a query will be retrieved first, followed by documents with words with lower correlations.

Associative message analysis is based largely on the work of Gerard Salton, at Cornell University, whose SMART (Salton's Magical Automatic Retrieval Technique) system has been extensively tested over the past ten years. The RADCOL system at RADC is the largest experimental implementation of Salton's associative analysis methods, and it serves as a unique tool for the testing of statistical approaches to natural-language processing.

2.3. ADVANTAGES AND DISADVANTAGES OF ASSOCIATIVE METHODS

Associative analysis has several advantages, in comparison with a keyword retrieval strategy:

- Rather than searching for the appearance of particular words in a data base, a query defines a general subject area, selecting the documents which are most relevant to this area of interest.

- When a document of interest has been located, it is possible to use this document itself as a query, retrieving other documents in the data base which deal with similar subject matter.

- Queries may be modified to emphasize areas of major interest, while retaining other areas of less interest as part of the query.

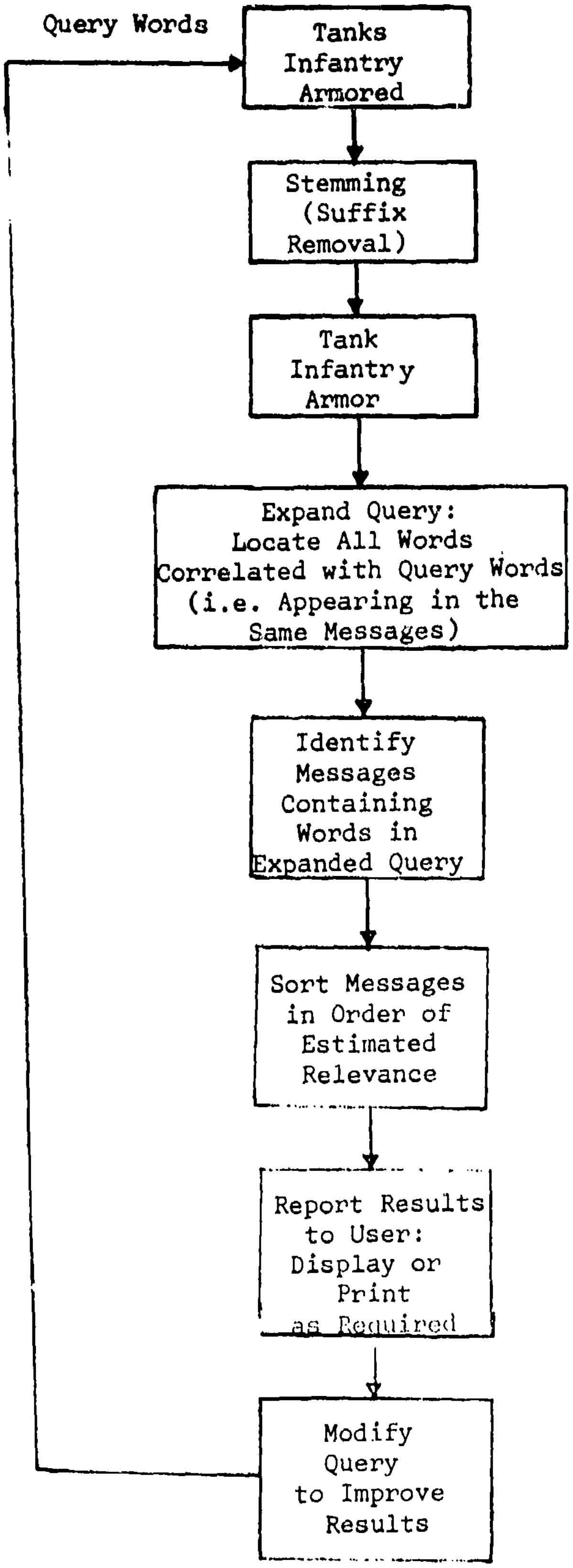


Figure 2-2 Associative Analysis

- An associative system constructs its own thesaurus, which is closely related to a data base. The correlations which appear in the thesaurus are those which actually occur in the data. The time-consuming task of manual thesaurus construction is not necessary.

On the other hand, a major disadvantage of associative analysis systems, like the RADCOL system, has been the excessive time required for initial data analysis, and for computation of correlations during interactive use of the system. A goal of the research reported here has been to reduce time and space requirements to acceptable levels, i.e., levels which are comparable to those of traditional keyword information retrieval systems. Initial feasibility studies indicate that this goal can be achieved.

2.4. RESULTS OF RADCOL STUDY

PAR has conducted an evaluation of RADC's initial implementation of an associative system, the RADCOL system, which utilized a 34,000-document subset of the Central Information Reference Control (CIRC) data base, consisting of abstracts of Science and Technology (S&T) intelligence reports. The system was later re-implemented with a simulated I&W intelligence data base, consisting of 1854 brief messages. A complete report of the RADCOL evaluation project is contained in RADC TR-75-208, August 1975.

In summary, results were as follows:

- Ability to locate highly relevant documents using unmodified queries, as originally prepared by the analyst in natural language, was very low.

- When these queries were modified by a system specialist, to remove unneeded words from the query, results were as follows:

  75 per cent -- highly relevant messages among the first five messages received

  90 per cent -- moderately relevant messages among the first five messages received

  10 per cent -- queries judged inappropriate for this type of system

- Further modification of the queries, by increasing or decreasing the weights of query terms, produced the following results:

  95 per cent -- highly relevant messages among the first five messages retrieved

  100 per cent -- moderately relevant messages among the first five messages retrieved

It was particularly encouraging to find that retrieval failures occurred for obvious and correctable reasons--the format of the messages, or a requirement on the part of the user that demanded other retrieval techniques. While a great deal of work remains to be done to improve system performance, these results clearly indicated the effectiveness of associative analysis.

Areas for possible improvement of system performance have also been recommended. In general, these are technical recommendations for reducing the time required for initial data processing and for reducing the computation needed to produce query-document correlations. PAR's studies have shown that it will be possible to reduce timings substantially.

In the initial versions of the RADCOL system (as delivered by another contractor to RADC), because of programming errors, analyses often required as much as two hours. Reports of this performance led to overestimates of retrieval times for associative systems. When errors were located and removed, timing was greatly improved. For initial search of an inverted list, which locates documents associated with the query, the system is as fast as any other system now available. Reports on the number of documents associated with a query are returned within five seconds. The correlation process, which is not performed by standard information retrieval systems, is time-consuming in the present implementation. In the time-sharing HIS 635 computer implementation, these correlations require from 30 to 90 seconds.

While these times are not unacceptable, it is likely that they can be improved in several ways, as summarized in Section 1.3.

In evaluating startup and update times, it is important to recall that associative methods eliminate much of the manual processing that has traditionally been required for the implementation of information systems. Data entry is in free-text form, without special preparation. Word lists and a thesaurus are generated automatically by the system, rather than by human coders. This automatic pre-processing greatly reduces overall startup time, by eliminating the most time-consuming part of keyword system implementation.

It is likely, therefore, that total system startup time for an associative analysis system may actually be less than the total startup time for a keyword retrieval system, which requires manual pre-preparation of the data.

With the successful demonstration of the feasibility of associative message processing techniques, it becomes possible to proceed with the design of an operational system for statistical message analysis. The capacity for locating documents which are relevant to a given subject area, rather than a simple search for keywords, represents a major extension of natural language processing capabilities.

2.5. DATA BASE PREPARATION

Data base preparation, in the present RADCOL implementation, is fully automatic. Statistical methods are used to select the vocabulary for retrieval, to perform correlations among the selected words, and to form the clusters or concepts which are used for actual retrievals. Several steps are required to prepare a given data base.

1. Word stems are derived by removing suffixes and other word endings.

2. A concordance or index of the locations of each of these word stems is prepared. The concordance lists the documents in which each stem appears, together with the number of appearances.

3. Stems are next ranked according to their potential value in discriminating among documents. This statistical filter eliminates stems which are not likely to be useful for retrieval.

4. The remaining stems, which form the retrieval vocabulary, are correlated to determine which stems tend to occur together.

5. In the current version of the RADCOL system, concepts or cluster centers are located. Each cluster of stems forms a concept, and these concepts constitute the basis for correlations between documents. Because of the time required for cluster formation, the poor quality of clusters formed with currently available methods on very large data bases, and the lack of a clear rationale for the use of clusters, this step is being eliminated. (A full discussion of cluster elimination is included in Section 3.7).

Documents in the data base are located and ranked according to the correlations between the words which they contain and the words which occur in the query. Such correlations act as a measure of similarity between the document and the query. With a variety of techniques, the user may obtain precisely those documents which lie within a designated subject area in the data base.

2.6. SYSTEM OPERATIONS

Specific system operations may be described in more detail as follows:

1. The initial RADCOL stemming algorithm was implemented on an empirical basis, by listing all words in the data base alphabetically by their endings--reversing the words, then sorting, and reversing again--and preparing a list of endings which were present in the CIRC data base. It was found that this list could be applied to the I&W data base. Additional stemming algorithms are being implemented and tested as part of the METER feasibility study, as described in Section 3.4.

2. Preparation of a concordance is a straightforward procedure, producing the word identifications that are required for the statistical studies which follow.

3. The statistical filter is one of the most interesting and critical operations of the system. The purpose of this process is to choose those word stems which are most effective in discriminating among the documents of the system. For example, we might expect that the word COMPUTER or the stem COMPUT would appear frequently in documents that deal with computation, but would not appear at all in most of the other documents in the data base. COMPUT would then be a good discriminator, since it would help to define a limited subset of the documents. On the other hand, some of the function words, like THE, AND, OF, and so on, might be expected to appear randomly throughout the data base. For this reason, they would not be regarded as good discriminators, since they would not serve to define any particular subset of the set of documents. In addition, words which appear only once or twice throughout the data base should be eliminated, since they do not represent any of the major subject classifications of the data. Many of the "words" which appear only once in the CIRC data base were found to be nothing more than typographical errors. In the I&W data base, latitude and longitude readings, times, azimuth readings, and other highly specific data which could not be effectively used for retrieval were eliminated through the use of the statistical filter. The statistical filter is discussed further in Sections 3.5 and 4.2.

4. Two stems are said to be correlated when they appear in the same documents. If they appear in exactly the same documents exactly the same proportionate number of times (i.e., in proportion to the length of the documents), their correlation will be 1.0. If they never appear in the same documents, their correlation is zero. (Alternative correlation measures are described in Sections 4.3 and 4.4). The vast majority of the pairs of stems in both data bases had correlations of zero, and a large proportion of correlations among the remaining stems tended to be very low. In fact, it is possible that the enormous numbers of very low correlations carried by the system were partially responsible for disappointing retrieval times with the CIRC data base. More than 90 per cent of computation time was spent with correlations which were less than 0.1, and which had no effect upon retrievals. By eliminating these low-level correlations, system performance can be greatly improved without loss of effectiveness.

5. The delivered version of the RADCOL system included a clustering routine, which has been eliminated from current system designs. It is described here for the sake of completeness.

The matrix of stem-to-stem correlations was the principal input to the clustering routines, which locate clusters or groups of stems which tend to occur together in the documents. The one-pass clustering routine, which PAR developed in re-implementing the RADCOL system, represented an approach to clustering which was feasible for a very large correlation matrix. It was not a clustering routine in the traditional sense, however. Instead of locating disjoint clusters, it located cluster centers, which are stems which tend to correlate highly with a particular group of stems. Each individual stem may be correlated with as many as four cluster centers, and may therefore be a member of more than one cluster; in a traditional clustering routine, each element belongs to only one cluster.

Although the clustering process was found to be unnecessary for retrieval functions, it did have a secondary benefit for the user of an information system, since it indicated the major areas of interest, or "concepts," that were present in the data. Words were gathered into clusters on the basis of the number of times they appeared together in the same documents. Each cluster represented a common area of interest or "concept" in the data base. By examining the clusters, the analyst could determine the types of subject-matter that were present in the data. Current METER system designs eliminate the clustering routines for reasons indicated in Section 3.7. Documents are retrieved when they contain word stems which are highly correlated with the word stems in the query; concepts or cluster centers are not used.

6. Following the correlation process, each document is represented by a list of the content stems (i.e., the word stems in the document which have passed the statistical filter described in step 3), weighted according to the number of times each word appears. These lists are called the Document Stem Vectors. In addition, there is a correlation matrix, showing correlations among all the content stems in the data base which exceed a given threshold (currently 0.3).

7. In normal operation, a query is first entered into the system. Words in the query are then stemmed, using the stemming procedure described at step 1. Word stems which are not contained in the system are reported back to the user. The content stems--i.e., the word stems which are contained in the system--are matched against the correlation matrix, and all stems having a significant correlation with the stems in the query are added to form the Query Stem Vector.

   All documents which contain word stems which appear in the query vector are located, and correlations are formed between the query vector and the document vectors. The highest correlations are retained and returned to the user. The document with the highest correlation is listed first, then the second-highest, and so on.

   Study of the RADCOL system, using an I&W data base, indicated that the correlations resembled human judgments of relevance to the extent that the most relevant documents are usually included among the documents with the highest correlation rankings.

8. Query modification. Salton's work has shown that the most valuable tool for improvement of associative analysis, which has given it an

advantage over keyword information retrieval, is the ability to modify the query concept vector to indicate precisely which concepts are desired. Two of the most useful methods, relevance feedback and direct concept vector modification, were not properly implemented for the RADCOL system as it was delivered to us, and it was therefore impossible to replicate Salton's results. However, it was possible to simulate these methods to some extent by adding and deleting words from the query, and by conducting document-document searches. Whenever the user finds a document which is relevant to his needs, he can use this document itself as a query to search out other similar documents from the data base. Query modification greatly improved retrieval quality, as noted in Section 2.4.

2.7. LIMITATIONS

The strengths of associative message analysis systems have been indicated; it is also important to be aware of potential shortcomings of the approach.

1. PAR's initial study of the RADCOL system indicated that clustering techniques were far from optimal. Since the present implementation, as modified to operate in a single pass, is very fast, and since the clusters which it forms are acceptable, the system could be continued in its present form. It was nevertheless found to be desirable to determine whether clustering could be eliminated without degrading system performance. Current work with RADCOL includes plans for eliminating the clustering procedure, and METER will operate without clustering.

2. A second capacity which would be valuable in the METER system would be the ability to search for specific terms in the data base. Since the system prepares a complete concordance, listing all stems that are found in the data, together with a reference to the documents in which these stems are found, it will be interesting to explore the possibility of making this list available to the on-line retrieval system, to permit the retrieval of documents containing specific words. Such an ability could be illustrated by locating documents containing the terms MIG-19 or FARMER, which were eliminated from the RADCOL I&W implementation by the statistical filter. Earlier, in the initial indexing of all word stems in the data base, MIG and FARMER had been indexed, but they were discarded when their statistical characteristics indicated that they would not be as useful as other words for retrieval. If the initial index could have been saved and searched, then it would permit retrieval of the required documents.

   Since the logic required for keyboard or Boolean retrievals is extremely simple, it would be possible to add Boolean retrievals (in which specific words or word stems are linked by AND, OR, and NOT) as an alternative to associative retrievals. This would permit a search for documents pertaining to specific terminology. Further suggestions for system expansion appear in Section 3.10.

3. Because associative retrieval is intended for free-text searches, no attempt has been made to use the formatted data which appear in the I&W and CIRC data bases. Both data bases include a great deal of header information which would be valuable for certain types of searches, and a full implementation will make use of some of this information.

   Such a capability would depend heavily on the specific formats contained in the data bases, but would normally include searches by author, title, date, and source. Geographical information is a valuable type of data included in the I&W implementation, and key words, intended for retrieval, are included in the CIRC headers.

   The usefulness of this option would be greatly increased if the user could specify dates or times with BETWEEN, GREATER THAN, LESS THAN, LATEST, or EARLIEST operators. For example, it should be possible to restrict retrieval to documents with dates between two limits. This would represent an extension of RADCOL's present ability to exclude documents with specified accession numbers.

4. Although the time required for system startup has been substantially reduced, there is still room for improvement in system update time. A major goal of the present design effort will be the development of algorithms for effective daily updates of the system.

   For this purpose, redesign of the statistical filter, development of more effective methods for forming stem-stem correlations, elimination of the clustering routines, and many additional design modifications will make daily updates possible. In addition, outputs from daily update routines will be of interest in themselves, since they will show changes in the thesaurus that occur through time. These changes would assist in detecting newly developing trends and patterns in the particular data base under consideration.

SECTION 3

RESULTS OF THE INITIAL FEASIBILITY STUDIES

In this section, current experimentation is described, on the basis of which system feasibility has been determined. A variety of different experiments have been performed, but the primary goal of this work has been to establish the software required for further testing of experimental algorithms.

Figures 3-1, 3-2, and 3-3 present an overview of the principal components of the proposed METER system.

3.1. CURRENT DATA BASE IMPLEMENTATIONS

The following data bases have been implemented and are available for testing:

1. I&W Message Data Base. A set of 1854 brief messages simulating I&W messages received during a fictitious crisis in the Middle East and during a fictitious Sino-Soviet border dispute, together with a substantial number of "noise" messages.

2. Vietnam News Reports, 1960-62. A collection of 1206 newspaper articles concentrating on the developing situation in the Far East.

Programs for initial analysis of these data bases have produced word frequency counts, counts and averages for word lengths, and other statistical information. Although statistical information will be essential for later tests and algorithm designs, it is not immediately relevant to the purposes of this report, and details are omitted here.

3.2. QUERIES FOR SYSTEM TESTING

A collection of queries was prepared, based on the Vietnam data base, to provide a set of tests for the METER system. The set of queries was intended to suggest the range of facilities which might reasonably be expected in the system. Since correct responses are known for all the queries, they will also serve to test system performance.

The following queries were prepared:

1. What was the role of helicopters in Vietnam during this period?

2. How effective was the strategic hamlets program?

3. How were dogs used in Vietnam?

4. What rifles were used by U.S., Vietnamese, and Vietcong troops?

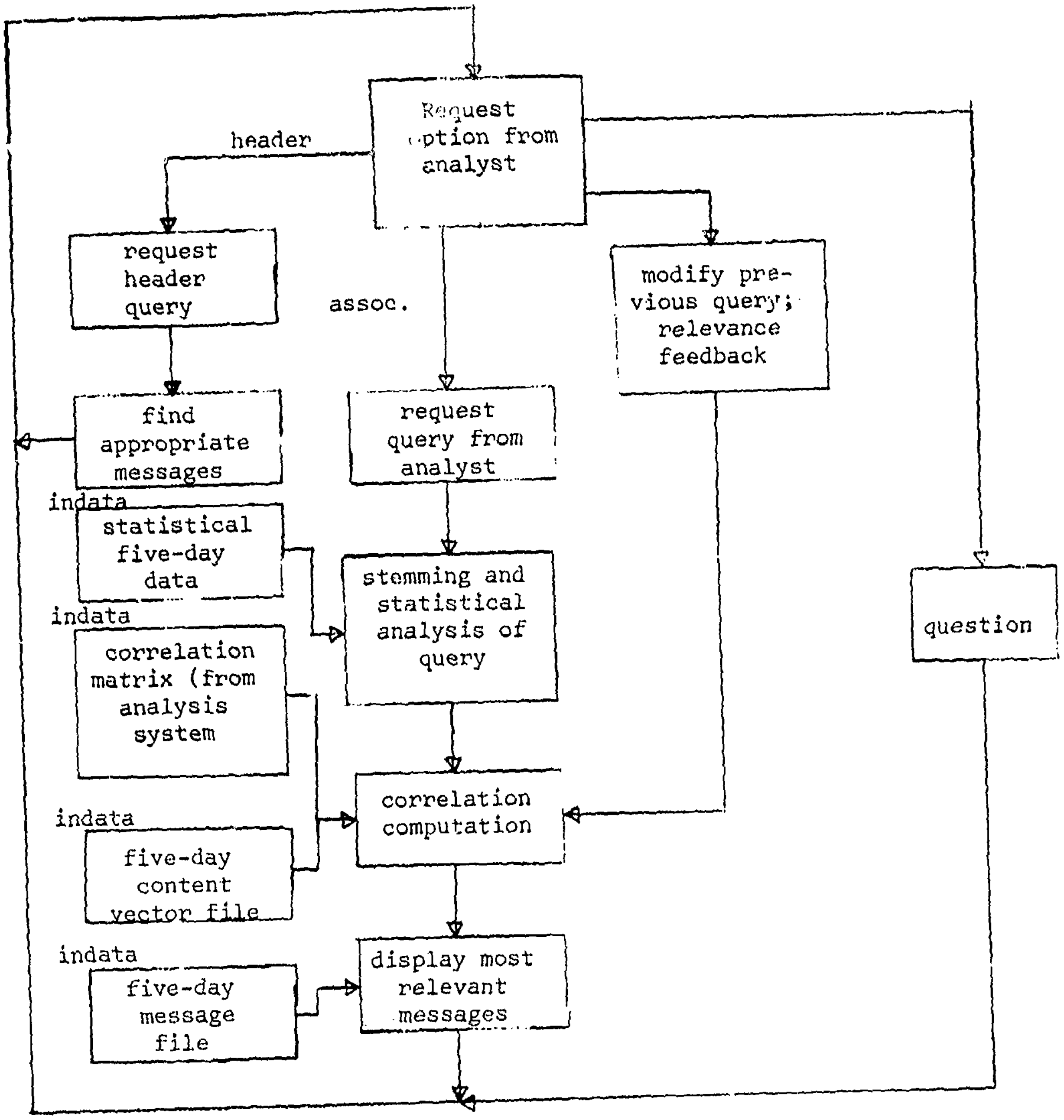


Figure 3-1 Statistical On-Line Message Extraction System

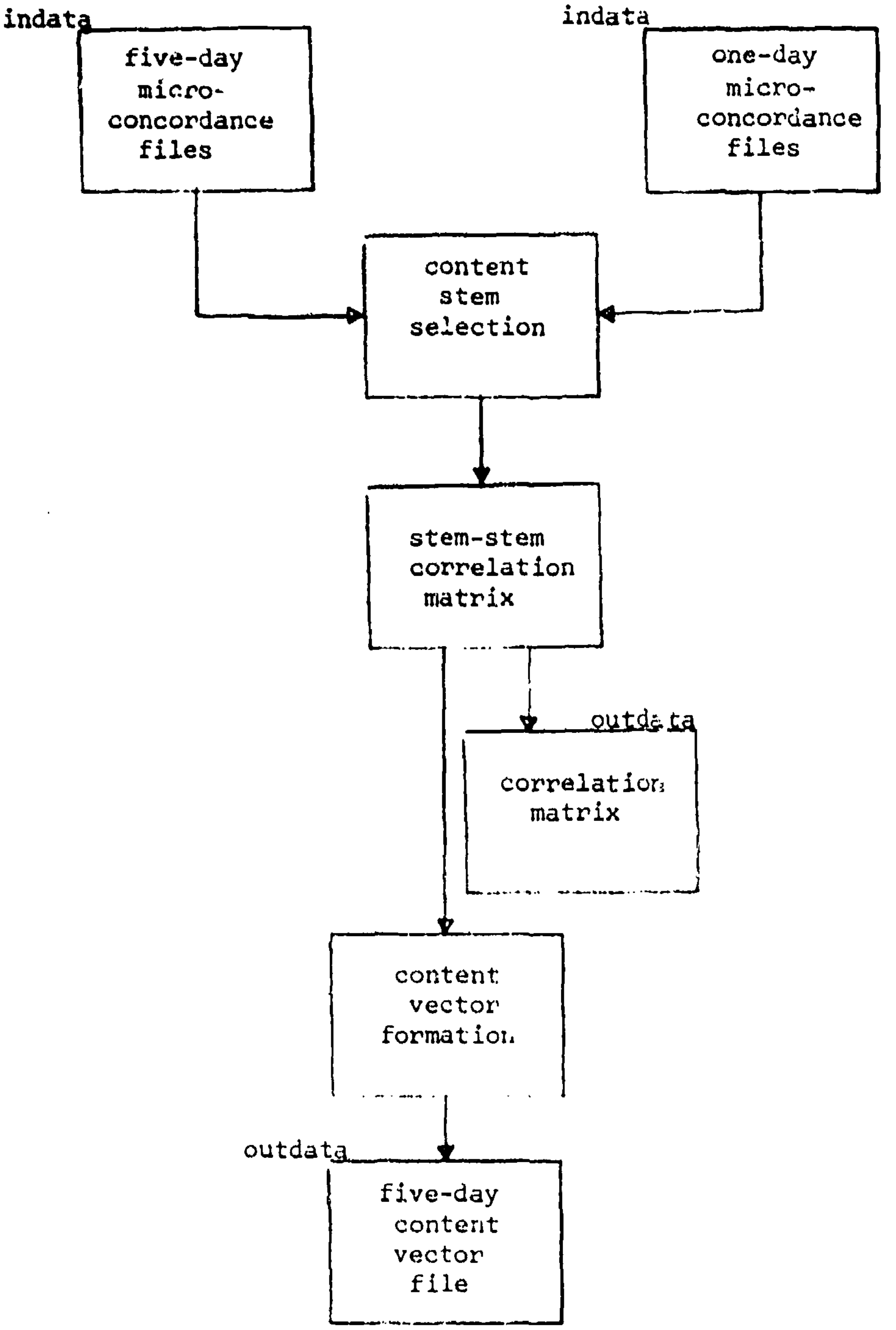

indata
five-day micro-concordance files
indata
one-day micro-concordance files
content stem selection
stem-stem correlation matrix
outdata
correlation matrix
content vector formation
outdata
five-day content vector file

Figure 3-2 Daily Associative System

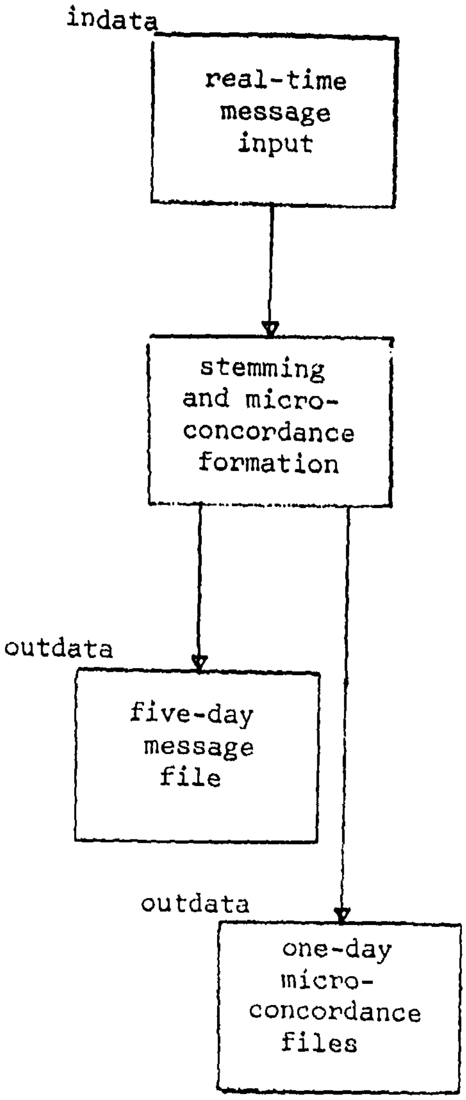

indata
real-time message input
stemming and micro-concordance formation
outdata
five-day message file
outdata
one-day micro-concordance files

Figure 3-3 Realtime Initial Processing System

5. Were electronic aids described? How were radar and radio used?

6. What was the role played by Buddhist suicides during this period?

7. Summarize as many as possible of the accidents, errors, and other non-combat-related injuries and deaths.

8. Trace the role of U.S. and Soviet technicians in the conflict.

9. Perform document-document searches on a selected set of articles.

10. Mrs. Nhu forbade dancing in Vietnam. How successful was her directive?

11. Agronomists were among the U.S. technicians. Trace their role, as well as that of other experts, technicians, and advisors.

12. How likely was it that the U.S. would introduce atomic/nuclear weapons? What was the role of uranium?

13. Engineering and technical aid is described in detail in two of the documents. Can these be retrieved?

14. What articles describe canal building?

15. There are several stories on terrorist bombings. Is it possible to determine how much effect terrorism had on the outcome of the conflict?

16. Was the U-2 reconnaissance plane used in Vietnam?

17. Describe the role of tanks in the conflict.

18. Locate background stories on the technology used for bridge building.

19. Were herbicides used for defoliation during this period?

20. Describe the role of chemical warfare in Vietnam.

The preceding queries will be used in later tests of system performance. They are included here to indicate the range of analyses which may be expected from an associative message system.

3.3. THE ALGORITHM DESIGN LANGUAGE, PASCAL

PASCAL is a block-structured algorithmic programming language based on ALGOL 60. Its major departures from ALGOL are a simplification of syntax for more efficient implementation and the addition of structured data types. Otherwise it resembles ALGUL closely and can be quickly learned by anyone familiar with ALGOL or PL/I. Like ALGOL, PASCAL allows a clean representation of algorithms and is easier to read than FORTRAN. It is compact enough to fit on machines as small as the PDP 8/E.

The development of the METER system will require writing a large number of programs dealing with non-numerical string data and with linked lists and trees. The currently available programming languages on the PDP 11/45, FORTRAN and MACRO-11, are both undesirable for this purpose.

FORTRAN has the disadvantage of having no provisions for structured data other than arrays. Operations such as string manipulation can be performed, but only in a clumsy way requiring programs that are longer and more complex than necessary. Furthermore, FORTRAN does not lend itself to top-down system design and implementation.

MACRO-11 assembly language provides a way of handling structured data, but tends to be laborious and error-prone. Programming can be facilitated through the use of macros, but the extent to which this is necessary for obtaining readable code almost amounts to the definition of a new programming language.

A higher-level language like PASCAL will meet these deficiencies. It will introduce standardization, make programming more concise, and enhance readability.

PASCAL is the first programming language designed from the start for structured programming. Its use should reduce development time and increase reliability. Since its application will be in the development of experimental systems, the planned implementation will include special instrumentation for measuring the efficiency of algorithms on the PDP 11/45.

It should be emphasized that PASCAL is not intended for use in production of the final product to be delivered as the prototype METER system; it will be used during the preliminary stages of the project for experimentation with many competing design strategies.

During the initial feasibility studies, a compiler for PASCAL was obtained, appropriate documentation was reviewed, and experimental programs in PASCAL were written. The compiler itself is written in PASCAL, and thus may easily be modified. Current plans call for a version of the compiler to generate PDP 11/45 assembly language as output, so as to make programs portable to non-PASCAL facilities.

3.4. STEMMING

The RADCOL stemming algorithm is based largely on a compilation of suffixes occuring within the CIRC data base, a bibliographic data base derived from S&T abstracts. Such an approach appears to be satisfactory when dealing with a static collection of documents, but it is unclear whether it will be adequate for the highly dynamic I&W environment, in which it is not known in advance what suffixes will occur.

An examination of the RADCOL algorithm as applied to the collection of simulated I&W messages indicated that:

- o Only about 10% of the words actually had suffixes removed.

- o Most of the suffixes removed fell into a small subset of all the possible suffixes.

- o Where a suffix was removed, the algorithm was likely to overstem.

Because of these shortcomings, it seemed desirable to investigate alternative stemming algorithms to determine how much performance was affected.

The following experimental alternatives have been studied:

- o No stemming. This establishes a benchmark for the effectiveness of stemming algorithms in the I&W environment, as simulated by available data bases.

- o Inflectional suffixes only. A few inflectional suffixes like S, ED, and ING appear to account for the majority of suffix occurrences in the simulated I&W data base. To deal with such suffixes, a highly sophisticated stemmer has been designed and is under test. This contrasts with the RADCOL approach, in which a simple stemmer works with many different suffixes.

- o A two-stage stemmer. It is possible to split stemming up into stages, e.g., by having an inflectional stemmer precede a more general stemmer. This is somewhat similar to the approach used by the RADCOL algorithm, which devotes a single pass to cutting off final S's; the inflectional stemmer simply extends this idea to other suffixes.

The stemming problem can be stated as follows:

The process of stemming involves the removal of prefixes and suffixes from a word in order to reduce it into a root form for the purpose of matching. For example, we might stem GIVEN and PASSING so that we can compare them against GIVE and PASS. The actual choice of prefixes and suffixes to be removed usually depends on the particular application. An algorithm which was satisfactory for an S&T data base might not be satisfactory for an I&W message analysis system, both because of differences in vocabulary and because of the different purposes for which these data bases are intended.

Two opposite types of hazard must be overcome by a stemming algorithm: over-stemming and under-stemming.

- Over-stemming occurs when two unrelated words are reduced to the same stem. For example, if APPLIES and APPLES were both reduced to APPL, then these two words would be confused, and the amount of noise in the system would increase.

- Under-stemming occurs when two related words are reduced to different stems. For example, if COMPUTERS and COMPUTING were retained by the system as distinct words, there would be no way of telling that they referred to closely related concepts. In addition, the vocabulary of the system would be needlessly increased.

In traditional terminology, over-stemming results in less precision (since unwanted words will be included in a retrieval), and under-stemming results in lowered recall (since wanted words will not be included).

The experimental stemming algorithm, in its current implementation, deals with inflectional suffixes, such as S, ED, and ING, since these are the most common suffixes, and since they contain syntactic information for possible use in language analysis. The inclusion of other inflectional suffixes like EN, ER, and EST and certain morphological suffixes like LY, FUL, MENT, and TION is being evaluated to determine whether the additional computation required is justified by a corresponding improvement in precision.

3.5. STATISTICAL FILTER COMPUTATION

When the collection of stems in the messages is reduced to a computationally feasible size, we need to make sure that we choose the stems which work best for retrieval. Our present basis for comparison is a measure of term importance developed by Sally Dennis, and hence called the Dennis Measure.

Other measures are being considered as alternatives, but this one is our first choice because it offers a potentially better response than the others. The Dennis Measure has been used in the RADCOL system.

We list the notation required for this section.

N is the number of documents,

F_t is the number of occurences of term t in the entire collection,

$f_{t,d}$ is the number of occurrences of term t in document d,

S_d is the number of terms in document d (each term is counted as often as it occurs),

$g_{t,d} = f_{t,d}/S_d$ is the "amount of document d" allocated to term t,

We consider, for each term t, the distribution over the documents of the g's. We need to know the mean and variance of this distribution.

$$\text{mean} \quad \bar{g}_t = \frac{1}{N} \sum_d g_{t,d}$$

$$\text{variance} \quad V_t = \frac{1}{N} \sum_d (g_{t,d} - \bar{g}_t)^2.$$

The Dennis measure C_t is defined by the following formulas:

$$e_t = \bar{g}_t^2/V_t$$

$$C_t = F_t/e_t = F_t V_t/\bar{g}_t^2$$

In order to compute the Dennis measure for our system, we need to have the equations in a different form. For convenience of notation, let

$$H_t = \sum_d g^2_{t,d}$$

$$G_t = \sum_d g_{t,d} = N\bar{g}_t .$$

Then

$$\begin{aligned}
NV_t &= \sum_d (g_{t,d} - \bar{g}_t)^2 \\
&= \sum_d g^2_{t,d} - 2 \sum_d g_{t,d}\,\bar{g}_t + \sum_d \bar{g}_t^2 \\
&= H_t - 2\bar{g}_t G_t + N\bar{g}^2{}_t \\
&= H_t - G^2{}_t/N, \quad \text{so}
\end{aligned}$$

$$\begin{aligned}
C_t &= F_t\, N^2 V_t/(N\bar{g}_t)^2 \\
&= F_t(NH_t - G_t{}^2)/\, G^2{}_t \\
&= \left(F_t \frac{NH_t}{G^2{}_t} - 1 \right)
\end{aligned}$$

This equation is our basis for further investigation.

Suppose we have a volatile message data base, i.e., one that is changing each day. We describe a procedure for computing the Dennis Measure for such a data base.

Assume that we are processing m days of messages.

Let F_{ti} be the number of occurrences of term t, i days ago (here $0 \leq i \leq m$, where i=0 denotes the new messages to be added), $G_{t,i}$ be the sum of $g_{t,d}$ for the documents d that were received in that day. $H_{t,i}$ the sum (over the same range) of $g^2_{t,d}$, and N_i the number of messages received that day. Then

$$F_t^{old} = F_{t1} + F_{t2} + \dots F_{tm}, \quad F_t^{new} = F_{t0} + F_{t1} + \dots + F_{t,m-1}$$

$$C_t^{old} = F_t^{old} \left(\frac{\sum_{i=1}^{m} Ni \quad \sum_{i=1}^{m} H_{t,i}}{\left(\sum_{i=1}^{m} G_{t,i}\right)^2} - 1 \right)$$

We may therefore compute new values as follows:

$$F_t^{new} = F_t^{old} + F_{t0} - F_{tm},$$

$$N^{new} = N^{old} + N_0 - N_m,$$

$$G_t^{new} = G_t^{old} + G_{t0} - G_{tm},$$

$$H_t^{new} = H_t^{old} + H_{t_0} - H_{tm}, \quad \text{and}$$

$$C_t^{new} = F_t^{new} \left(\frac{N^{new} H_t^{new}}{(G_t^{new})^2} - 1 \right)$$

There will be no division by zero if we remove all terms with $F_t^{new} = 0$ from consideration, since any term with $G_t^{new} = 0$ has every $g_{t,d}$ zero for all messages in the last m days, so that every $f_{t,d}$ is also zero.

This process requires storing the values $F_{t,i}$, $G_{t,i}$ and $H_{t,i}$ for $0 \leq i \leq m$ for each term, and also N_i for $0 \leq i \leq m$.

As a result of this study, we now have an efficient method for computing the Dennis measure for each day's messages, without requiring extensive recomputation. The development of this algorithm represents a significant achievement of the initial feasibility studies.

3.6. CORRELATION STUDIES

The correlation used by many earlier associative processing systems, such as the RADCOL system, represented the cosine of the angle between two vectors in a conceptual hyperspace, and thus was not a correlation in the statistical sense. It actually represented an estimate of the probability that two stems would occur in the same document, and thus it took on values only in the range from zero to one, rather than values from -1 to +1.

The cosine correlation behaves much like statistical correlation, but the deviation between the two increases with the decreasing probability of the stems being correlated.

Experiments are under way to determine the effect of various types of correlation measures upon analysis efficiency, and to determine whether the correlation measure interacts with the statistical filter described in Section 3.5.

These experiments are discussed in detail in Sections 4.2 and 4.3.

3.7. AN APPROACH TO CLUSTERING

Initial implementations of the RADCOL system required the formation of "clusters" or "concepts" among the words contained in the data base. Although the motivation for this procedure was not clearly stated, it appears to have served such purposes as the following:

- In some sense, it was thought that clusters would represent "concepts" or areas of interest in the data. For example, it was hypothesized that COMPUTER would appear frequently with DIGITAL, but rarely with HELICOPTER.

  Clusters or groups of words might be located, consisting of all those words which tended to appear more frequently with other words in the cluster than with words outside the cluster. COMPUTER, DIGITAL, DATA, etc., would appear in one cluster; in another, HELICOPTER, AIRCRAFT, ROTATING, WING, etc., would appear. Each of these groups of words would rerresent a "concept" or subject area.

  If it was therefore possible to locate a limited number of concepts in the data base, then documents could be indexed according to the set of concepts which they contained. This automatic index would give the location of specific concepts in the documents comprising the data base.

- A study of the clusters in the data base could be of interest to an analyst, since it would show specific concept areas which appeared there. For the current intelligence analyst, such information would be particularly important, because it could be used to detect developing concept areas. For example, the appearance of a cluster of words such as ARMS, ARMAMENT, and ANGOLA might provide an early warning of potential trouble.

- Another reason for including the clustering algorithm would have been the desire to reduce storage requirements for document analysis. If a document were found to contain ten concepts, for example, then only ten numbers would have to be stored; and only these ten concepts would be needed to form correlations with queries or with other documents. In addition, only a limited number of concepts would be stored, with a resulting saving in memory requirements.

Experimentation with the RADCOL system indicated that clustering, as implemented there, was not sufficiently advantageous to outweigh its serious shortcomings. Specifically:

- Effective clustering algorithms require extensive manipulation of the data, which becomes infeasible when the number of data items to be clustered is large. Typically, some 4,000 to 5,000 word stems are used.

  Similarly, standard factor analysis techniques are inapplicable to data bases of this size, even on the dubious assumption that the data meet the statistical requirements for factor analysis.

  The clustering algorithm provided with the RADCOL system depended heavily on chance for the designation of cluster centers. As a result, while the algorithm could be executed within a reasonable time, the clusters were frequently found to be badly formed.

- Empirical investigation of the clusters formed by the RADCOL system did not provide information of interest. There was little that could be determined about the character of the messages in the I&W data base from an inspection of the clusters.

- Although the quality of searches performed by the RADCOL system was very good, there is no indication that this quality might not be maintained with an unclustered approach, which was given the name "Direct Linkage Algorithm."

- A study of storage requirements indicated that the clustering procedure used by the RADCOL system did not provide a substantial saving in memory requirements, as compared with the proposed Direct Linkage Algorithm.

For the preceding reasons, it was felt desirable to investigate a method of by-passing the clustering routines, through the Direct Linkage Algorithm. It seemed likely that this approach would provide all the advantages that could have been claimed for a clustering algorithm, without the shortcomings noted above.

In particular, it was felt that outputs could be designed in such a way as to permit the investigation of high correlations in the data, on a daily basis, which would give the analyst information about the changing patterns of events in the day's messages, without the need for cluster formation.

In the discussion which follows, it should be noted that "stems" refers to content stems, i.e., words from which the suffixes have been removed and which have been statistically selected for their ability to discriminate among the documents in the data base.

A "document stem vector" is simply a list, for each message, of the stems contained in it, together with a count of the number of times each stem appears.

A "query stem vector" is a similar list of the content stems contained in the query. When document-document searching is to be performed, then the appropriate document stem vector is used as a query stem vector.

Finally, the stem-stem correlation matrix is required. This is estimated at approximately 5000 x 5000 in size, with 99.9 per cent of the cells empty, when a threshold of 0.2 is used to eliminate small correlations.

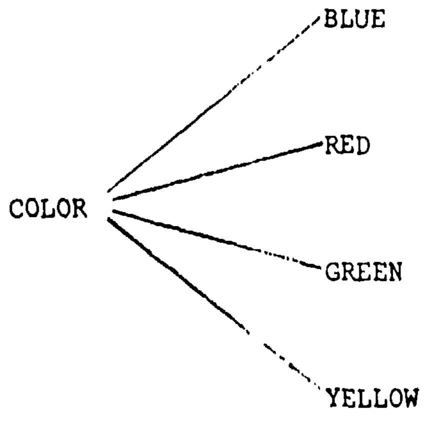
Searches proceed in the following way. The Query Stem Vector is formed. This is then expanded by multiplying each stem by a weight corresponding to the number of times that it appears in the message, divided by the length of the message. For each stem in the message, all stems with a correlation greater than a given threshold are located in the stem-stem correlation matrix. Each correlation is multiplied by the appropriate weight, to obtain the Expanded Query Vector.

Finally, each Expanded Query Vector is correlated with each Document Stem Vector to obtain a correlation coefficient for the relationship between the query and the message.

A computationally effective method for performing the above procedure represents one of the major research goals of the initial feasibility study for the METER system. This goal has been achieved.

3.8. SECOND ORDER CORRELATIONS

The preceding description of the correlation process gives what we will call "first order correlations." Each word in the query is expanded to include all words with which it is correlated. This process can be illustrated as follows. Suppose that the query contains the word COLOR:

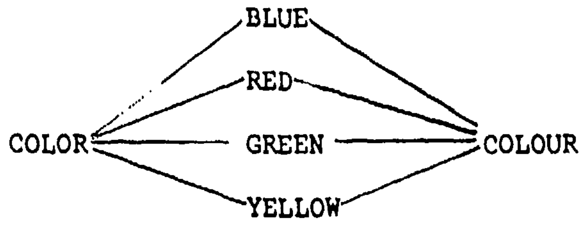


The system will search for messages containing the stems BLUE, RED, GREEN, and YELLOW, since the word COLOR has a non-zero correlation with these. It will also search for messages containing the word COLOR, since each stem has a correlation of 1.0 with itself.

However, when only first-order correlations are used, the American spelling COLOR is not likely to correlate with the British spelling COLOUR. Thus, the word COLOR in a query is not likely to retrieve any messages containing the word COLOUR.

More generally, in short messages, synonymous words are not likely to be correlated, since the message-writer will use only one word with a given meaning, not both.

To retrieve synonyms, therefore, another approach is required. Second order correlations are used in the following way:



It is probable that both COLOR and COLOUR are highly correlated with such words as BLUE, RED, GREEN, and YELLOW. When COLOR is entered as a query word, we search for messages containing such stems as BLUE, RED, GREEN and YELLOW. But we also search for the stems with which these are correlated, such as COLOUR.

The procedure described here is quite general. It is expected that second-order correlation will be successful in locating pairs of synonymous stems, since synonyms are likely to occur in the same contexts.

It should be noted that the use of second order correlations will provide all retrievals which could be obtained through the use of clustering procedures, since the intermediate links in the chain of correlations are used in exactly the same way that the cluster centers or concepts are used in the early versions of the RADCOL system. Essentially, all stems are used as cluster centers.

At this point, experimentation has not proceeded far enough to determine the effectiveness of the second order retrievals. Empirical data will be required to find out whether second order correlations tend to converge on valuable synonyms, or whether they tend to diverge, without adding any words that will be useful for retrieval.

A further discussion of mathematical methods is included in Section 4.2.

SECTION 4

THEORETICAL DISCUSSIONS

The discussions contained in this section include several position papers, which were prepared by members of the project team.

The following topics are discussed:

- Representation of Meaning
- Probabilistic Methods
- A Linguistic Interpretation of Interword Correlations in Text
- Use of the Correlation Measures
- Analysis of a Square Root Algorithm
- Query Formation
- Optional Additions to the Basic System

4.1. REPRESENTATION OF MEANING

We want to represent the meaning of a document in terms of easily isolated features. The requirement of ease of isolation is introduced to keep programs from becoming prohibitively long (in space or in time): the reason for using a computer in the first place is that the data bases are getting out of hand.

At present, the features chosen for a document or query are the content stems (i.e, the word stems which have passed the statistical filter), together with their frequencies. Moreover, the choice of content stems is made only on the basis of the raw stems and their frequencies. One problem with using stems as the only features can be illustrated as follows: consider messages containing the detailed description of two photographs of unrelated objects. The words used to describe the photographic images will be similar (words describing such features as color, size, and so on), even though the objects are unrelated. The two photograph descriptions would then erroneously be highly correlated. We therefore need a method for determining something of the content of a document, independent of the words used.

Another difficulty in using only these features is one of absolute performance (i.e., not of speed, but of accuracy). The information retrieval programs currently in existence seem to retrieve irrelevant documents quite often (poor precision) and miss relevant documents as well (poor recall). There is within each program a window through which the documents of the data base must pass in order to be retrieved. Most of the programs can adjust the location of the window either in the direction of retrieving more of the relevant documents (higher recall), or in the direction of retrieving fewer of the irrelevant documents (higher precision).

We will try to do both by noting that all information retrieval programs begin by throwing away most of the information in each document. If we compare the performance of a program with the performance of a person who can only read the list of content stems of a document and their frequencies, then the discrepancy might not be as large as it is when the person can read the entire document.

Only a few experiments have been performed which tried to examine other possible features, and none with large data bases. An example of such a feature would be to take pairs of stems. This kind of feature, i.e., a phrase or context feature, has the advantage of being relatively easy to recognize. It also illustrates an unfortunate property of these features: there are usually too many of them. For example, if there are 6000 stems in the data base (a small number, as compared with our problem), there are almost 18 million stem-pairs.

In order to determine which features of a document are most helpful for this problem, we need to consider the different levels of information that occur in a message, e.g, designation of objects at the lowest level, description at a middle level, and communication of purpose at the highest level. Ideally, there should be comparisons of performance made at each level in order to determine the significance of information at that level.

One way to effect this comparison is to have two people at a terminal, looking at the simplified documents, and deciding which are relevant to the question. Then whenever there is disagreement on the relevance of a document, the person with more information should describe what extra information he used to make his decision.

One approach to the problem of extracting a maximum amount of information from the documents is to parse the sentences in them, using any of the natural language processors presently available. We are developing several such processors in PAR's Language Laboratory (a version of LINGOL is being implemented on the PDP 11/45 at PAR offices in Rome, and other language analysis and design tools are being developed.)

A major difficulty with the naive algorithms for syntactic processing of documents by parsing is that each word needs to be in a dictionary in order to guarantee proper parsing (and even then there is no such guarantee); the processors need parts of speech or definitions or some classification of words. It is possible to make fairly accurate determinations of parts of speech from morphological content and syntactic position as established by the occurrence of function words; but this fails for terse or highly formatted messages.

A solution to this problem may lie in the following simple observation: We do not need to find out what the meaning of a sentence (or a message) is in conventional terms. We only need to be able to compare meanings as being similar or not. Nobody seems to know how to do this yet.

A more difficult problem is to try to consider the incoming messages as descriptions of external events and to relate each message to a collection of background information in order to determine its significance. For this purpose, we need some kind of representation of useful background information, as well as a comprehensive compilation of such information. Among other things, the information available would have to describe the basic structure of the I&W environment. This also seems to be difficult.

A minimum feasible level is the statistical level, where a message is abstracted into a vector consisting of its term-frequency values. This approach will be described in Section 4.2, since it comprises what is computationally feasible for this problem at this time. Even though most of the information in each message is ignored, this level has been found to be sufficient for fair message extraction from small data bases, as Salton's studies have shown, and from one large data base, in the RADCOL development. More study of the theoretical aspects of message analysis is warranted. As the theoretical problems are solved, they will be examined for computational feasibility.

A problem with the low statistical level of analysis is that no account is taken of the interrelations between the terms. Somewhat more information can be extracted by considering the document vectors to be in a space of meanings. Each term then corresponds to a directed axis in the space. Synonyms and antonyms should correspond to lines which are nearly parallel. For synonyms, the axes will point in the same direction (we will say in this case that the axes are parallel). For antonyms, the axes will point in nearly the opposite direction (in this case, the axes are anti-parallel).

Any relative measure of synonymity can be used. One way would be to get the measure externally, i.e., have a numerical thesaurus available. The drawback with this procedure is that obtaining all of the terms that might occur in the messages requires time, and keeping them requires storage space. Furthermore, the sense of a term such as "plant" varies with its context of use.

Another possibility is the automatic computation of synonymy, using the available data bases. Two words which have similar meanings will tend to occur in the same contexts, and two words which have independent meanings will tend to have dissimilar contexts. A problem then arises in trying to determine a computationally useful definition of the context of a term. In the minimum information case, the best we can do is to use the entire document containing a term as the context of the term. If more detailed processing is performed, the context may be localized to the appropriate sentence, or even phrase segment.

A more extreme view results in the definition of the meaning of a term to be the collection of all of its possible contexts. The meaning of the individual terms can be built up from a document collection as before, at the cost (again) of enormous quantities of storage space. For fairly small data bases, however, this approach is quite reasonable (see Lewis, et al., JACM 14 (1967) for a primitive investigation of this argument).

Another difficulty with this definition of meaning (also pointed out by Lewis, et al.) is that antonyms will also tend to occur in the same contexts. For example, with a purely statistical characterization of context, one cannot distinguish different distributions of negation in a text, and so one can systematically replace a term "x" with "not y," where y is the antonym of x. There does not seem to be an easy way around this problem without some external information being supplied to the programs.

This problem may not be so serious. It seems that the use of a term in a message, and the use of its antonym, will be along the same semantic dimension (e.g., size, color, or brightness). In this light, the inability to distinguish between synonyms and antonyms may only reflect our inability to discriminate among meanings beyond broad semantic categories. A possible solution to this problem would be to have a table of antonyms specifically for the semantic dimensions relevant to a particular application.

4.2. PROBABILISTIC METHODS

Since it has been shown in the evaluation of the RADCOL system that probabilistic or statistical methods can give reasonable results, there remains the problem of deciding what kind of statistics to use. There is no theoretical basis for preference of one method over another, since there is as yet no good theoretical description of the meaning of language elements. We must therefore proceed empirically, with perhaps some intuitive arguments for "why this should work" (an example of this kind of argument is given at the end of this section).

We will concentrate on the calculations that depend only on the term occurrences $f_{t,d}$, or on quantities derived from these numbers.

There are still two areas of experimentation. We need to find a measure m(t,u) of how synonymous two terms t and u are (this phrase should be interpreted properly for terms that may not be stemmed words, if any are used as features). We also need to decide how these measures should affect the relevance of one message to another. This latter question will be deferred to the next section.

For example, we can consider each term as a random variable, and each document as a sample value. Then a measure of similarity between terms t and u can be given by a normalized correlation, C_{tu} as defined below.

We recall that the number of documents is denoted by N, the number of occurrences of term t is F_t, the number of occurences of term t in document d is $f_{t,d}$.

If we denote the mean and variance of the random variable t by m_t and W_t, respectively, then the equations we use are derived next.

$$m_t = E(t) = \sum_d f_{t,d}/N = F_t/N,$$

$$m_u = E(u) = \sum_d f_{u,d}/N = F_u/N,$$

$$W_t = E(t^2) - E(t)^2 = (\sum_d f_{t,d}{}^2 N - F_t{}^2)/N^2,$$

$$W_u = E(u^2) - E(u)^2 = (\sum_d f_{u,d}{}^2 N - F_u{}^2)/N^2, \text{ and}$$

$$C_{tu} = \sum_d (f_{t,d} - m_t)(f_{u,d} - m_u)/ W_t W_u.$$

It may be more reasonable to use the values $g_{t,d}$ instead of $f_{t,d}$, where we recall that by definition $g_{t,d} = f_{t,d}/S_d$, and S_d is the number (counting multiplicity) of terms in document d, i.e., $S_d = \sum_t f_{t,d}$.

Another major candidate for our term-term measure is the normalized inner product, which we describe below for the case in which the numbers $f_{t,d}$ are used. We denote the length of the "vector" for term t by ℓ_t, and the standard deviation of the random sample given by the documents by σ_t. We also denote the normalized frequency values by $n_{t,d}$,

$$\ell_t^2 = \sum_d f_{t,d}^2$$

$$\sigma_t^2 = \frac{1}{N} \sum_d (f_{t,d} - m_t)^2$$

$$n_{t,d} = (f_{t,d} - m_t)/\sigma_t.$$

Then the correlation is

$$C_{tu} = (\sum_d n_{t,d} n_{u,d})/N,$$

and the normalized inner product is

$$P_{tu} = (\sum_d f_{t,d} f_{u,d})/\ell_t \ell_u.$$

We can compare these values for certain kinds of terms as follows:

$$N \sigma_t \sigma_u C_{tu} = \sum_d (f_{t,d} - m_t)(f_{u,d} - m_u)$$

$$= \sum_d f_{t,d} f_{u,d} - \sum_d f_{t,d} m_u - \sum_d f_{u,d} m_t + \sum_d m_t m_u$$

$$= \ell_t \ell_u P_{tu} - F_t m_u - F_u m_t + N m_t m_u$$

$$= \ell_t \ell_u P_{tu} - F_t F_u/N,$$

since

$$F_t = Nm_t \text{ and } F_u = Nm_u.$$

Also,

$$N\sigma_t^2 = \sum_d (f_{t,d} - m_t)^2 = \sum_d f_{t,d}^2 - 2 \sum_d f_{t,d} m_t + \sum_d m_t^2$$

$$= \ell_t^2 - 2F_t m_t + Nm_t^2$$

$$= \ell_t^2 - F_t^2/N.$$

If a term t is infrequent, then F_t/N will be relatively small, so F_t^2/N^2 will be near zero. Then we see that σ_t is about $\ell_t/\sqrt{N}$. Furthermore, if term u is not too frequent (the words with very high frequency will tend to be stop words, i.e., useless words such as "the", "and", etc.), then $F_t F_u/N^2$ will also be near zero, so that we get $\sigma_t \sigma_u C_{tu}$ is near $\ell_t {}^\wedge{}_u P_{tu}/N$ and then C_{tu} is near P_{tu}.

The choice of which of these two measures to use is complicated by two considerations. The first is that the normalized inner product is a little easier to compute. The second is that the correlation makes more sense linguistically (see the argument at the end of this section). The final choice must therefore await experiments with real data.

If we use the g's instead of the f's, then we can make a comparison between C and P under an assumption on the Dennis measure, (See Section 3.5.). We recall that

$$G_t = \sum_d g_{t,d} \quad \text{and} \quad H_t = \sum_d g_{t,d}{}^2$$

and we have used V_t for the variance and $\bar{g}_t$ for the mean. Then

$$NV_t = H_t - G_t{}^2/N,$$
$$V_t = H_t/N - (G_t/N)^2,$$

$$N\sqrt{V_tV_u}\ C_{tu} = \sqrt{H_tH_u}\ P_{tu} - G_tG_u/N,$$

$$(V_tV_u)^{1/2}\ C_{tu} = (H_tH_u/N^2)^{1/2}\ P_{tu} - (G_tG_u/N^2).$$

Now $C_t = F_tV_tN^2/G_t{}^2$ is the Dennis measure, so we are most interested in this analysis when C_t is large, say that

$C_t > \lambda N$ for some $\lambda > 1$ (See Section 3.5.) Equivalently,
$F_tV_tN > \lambda G_t{}^2$, or
$\bar{g}_tV_t > \lambda (G_t/N)^2$.

If the term t does not occur too frequently, then $\bar{g}_t$ will not be much more than 1, and will possibly be much less. Since each $g_{t,d}$ is between 0 and 1 (in fact, almost always much less), we have V_t much less than 1.

Finally, we note that these assumptions imply that $(G_t/N)^2$ is much less than 1, so that V_t is near H_t/N, and similarly, V_u will be near H_u/N, so that C_{tu} is near P_{tu}.

This argument seems to indicate that for important terms (as determined by the Dennis measure), these two measures of closeness of meaning will give the same results. It must be emphasized, however, that due to the large number of assumptions we have made, a final choice must wait for experimental verification.

We might be able to get some theoretical results here by a detailed examination of some real data. In particular, under the present arrangement, we will be using several thousand content terms, and the precise choice of this number will determine the parameter λ. It is expected to be more than 1 (a term t which occurs in exactly one document with frequency b has a Dennis measure $C_t = b(N-1)$, and variance $V_t = b^2(N-1)/N^2$).

A different measure of similarity has been called the Tanimoto measure:

$$T_{tu} = (\ell_t \ell_u P_{tu})/(\ell_t^2 + \ell_u^2 - \ell_t \ell_u P_{tu})$$

$$= P_{tu}/(s - P_{tu}), \text{ where}$$

$s = \ell_t/\ell_u + \ell_u/\ell_t \geq 2$ (recall that since we are only making these computations for terms that do occur, we have $\ell_t \neq 0, \ell_u \neq 0$). As the normalized inner product P_{tu} varies from 0 to 1, the Tanimoto measure

T_{tu} varies from 0 to a number less than or equal to 1 (and equality can only occur when $\ell_t = \ell_u$).

Another kind of measure of closeness of terms t and u can be computed as the excess of the actual number of documents in which t and u both occur over the expected number of such documents assuming random distribution (so that the number will be zero if the two terms occur independently). In order to normalize this value to a probability, we may divide by the total number of documents.

If we write $h_{t,d} = 1$ if term t occurs in document d, and $h_{t,d} = 0$ if term t does not occur in document d, then the numbers above are derived as follows: here only, we let H be the number of documents in which t occurs, so

$$H_t = \Sigma_d h_{t,d},$$

and the number of documents in which both t and u occur is

$$\Sigma_d h_{t,d} h_{u,d},$$

so the probability mentioned above is

$$X_{tu} = (\Sigma_d g_{t,d} h_{u,d} - H_t H_u/N)/N$$

$$= \Sigma_d h_{t,d} h_{u,d}/N - (H_t H_u/N^2),$$

so this measure is to the binary document vectors (the h's) as the correlation C_{tu} is to the weighted document vectors, whereas the measure corresponding to the product P_{tu} is the frequency of co-occurrence

$$K_{tu} = \sum_d h_{t,d} h_{u,d} / N.$$

One last measure will be mentioned, since it also involves binary vectors. It is vaguely like the Tanimoto measure, but not as closely related to earlier measures as the previous binary measures are. It has been used by (Maron & Kuhns JACM7 p. 216 (1960)).

For a binary vector e, write $\bar{e}$ for the complementary vector (each coordinate complemented). Then write $e \cdot f$ for the usual dot product of vectors. Write h_t for the vector whose components are $h_{t,d}$. Then this last measure is

$$M_{tu} = \frac{(h_t \cdot h_u)(\bar{h}_t \cdot \bar{h}_u) - (h_t \cdot \bar{h}_u)(\bar{h}_t \cdot h_u)}{(h_t \cdot h_u)(\bar{h}_t \cdot \bar{h}_u) + (h_t \cdot \bar{h}_u)(\bar{h}_t \cdot h_u)} ,$$

where some simplification may be gained by noting that

$$(h_t \cdot h_u) + (h_t \cdot \bar{h}_u) = H_t,$$

$$(h_t \cdot h_u) + (\bar{h}_t \cdot h_u) = H_u,$$

$$(\bar{h}_t \cdot \bar{h}_u) + (h_t \cdot \bar{h}_u) = N\text{-}H_t,$$

$$(\bar{h}_t \cdot \bar{h}_u) + (\bar{h}_t \cdot h_u) = N\text{-}H_t,$$

which identities follow from considering binary vectors as sets of documents. The value of this measure is always between -1 and 1 (these values occur when the terms in the numerator have one side or the other 0; both sides cannot be 0).

4.3. A LINGUISTIC INTERPRETATION OF INTERWORD CORRELATIONS IN TEXT

The statistical correlation of content words in a collection of documents is theoretically significant from a linguistic standpoint because it provides a quantitative characterization of the contexts in which words occur. Such information is important since the semantics of a particular word are completely defined by an enumeration of its contexts of occurrence. Correlations between words thus capture something of their structurally determined meanings and so would be useful in text processing applications where one is concerned with determining the content of texts.

The problem is in the choice of correlation measure. Since different measures can have different linguistic relevance, one must be careful to select the right one for a particular application. As an illustration, it is useful to compare a standard product-moment correlation versus a normalized dot product correlation on the ability to make semantic distinctions in an associative message processing system.

The normalized dot product correlation is defined as follows

$$P_{tu} = \frac{1}{\ell_t \ell_u} \Sigma_d f_{t,d} f_{u,d}$$

where t and u are the words being correlated and d is an index variable for documents. In this computation, each word W is treated as vector in N-dimensional document space having components fw,d equal to the frequency of occurrences for word w in document d. The correlation is simply the scalar product of the vectors for t and u with normalization by dividing by the lengths of the two vectors, ℓ_t and ℓ_u. Since each $f_{w,d}$ counts occurrences, the values of the $f_{w,d}$ are non-negative and $P_{t,u}$ is monotonically increasing with d.

The product moment correlation is on the other hand

$$C_{tu} = \frac{1}{N} \sum_d n_{t,d}\, n_{u,d}$$

where t, u, and d are as before and where N is the total number of documents and the $n_{w,d}$ are the frequencies of occurrence for word w in document d normalized to a zero mean and a unit standard deviation. This is the more familiar correlation measure, taking values between -1 and +1 here.

The two measures are computed in a similar way; the key difference is that one allows negative values of correlation while the other does not. This is significant, because of how these correlations will actually be used. Consider the following two hypothetical documents

(d) ...large...unusual...leafy...plant...productive...machinery

(d') ...large...unusual...manufacturing...plant...productive...machinery

Suppose that we not wish to retrieve documents about "unusual agricultural crops." Since a relevant document such as d may not actually contain "agriculture" or "crop," one typically expands a query by also looking for documents containing words correlated with the words in the query. For example, this might include "productive" and "plant," thus allowing a match with d. The only trouble is that this also retrieves d', an irrelevant document.

We can improve the discrimination of our query with negative correlations. If we know that "agriculture" is negatively correlated with "manufacturing," then we can downgrade the relevance of d' with respect to our original query. Note that this is not possible with only non-negative correlation values; the lowest possible correlation is zero, which would have only the effect of ignoring the occurrence of "manufacturing" in d'.

Although the two correlation measures described here are both statistically valid, it makes a difference in text processing which one we use. The product moment correlation seems to be more linguistically precise, and since it does not require much more computation, it would seem to be better choice here.

4.4. USING THE MEASURES

In this section, we suppose that we have computed a closeness measure m(t,u) for each pair of content terms. For notational convenience, we will denote the corresponding matrix by R (though we may use C, P, T for f's, g's, h's, or X, K, or M). These measures will always give values between -1 and 1, and often between 0 and 1 (for P, T and K: the others can have negative values).

There are several ways to use the relation matrix R. We will restrict ourselves to those uses which have some intuitive justification. There are linguistic reasons for choosing a particular geometrical interpretation of R, and several kinds of distances to use for each geometry.

We call the entry R_{tu} for terms t and u the first-order relation between t and u, $(R^2)_{tu}$, the (t,u) entry of the matrix RR, the second-order relation, and in general $(R^n)_{tu}$ the nth-order relation.

For each of our measures of relation, a high first-order value indicates frequent co-occurrence, and a low value indicates disjoint occurrence (the two terms never occur together). Thus two terms which have a large first-order relation are often in each other's context. It follows that a large second-order relation indicates a similarity of meaning, since the words tend to have the same context (the terms occurring often in the context of term t match the terms occurring often in the context of term u). Therefore, if we want to consider a measure of similarity of meaning, we should use R^2.

We point out that the term "similarity of meaning" can be interpreted as synonymy only for those measures of relation which can take on negative values. Then two synonyms give values near 1 and two antonyms give values near -1. For those measures which are never negative, we have values near 1 for synonyms as well as for antonyms (antonyms will also tend to occur in the same contexts).

We note that the t row of the matrix R measures the context of term t, so that a comparison of similarity of context can be obtained by using any of the similarity measures from before on two rows of R. Then we should replace R by a new matrix whose t,u entry is the similarity between the vectors given by row t of R and row u of R.

Let us now turn to a definite interpretation of our remarks, and give suitable definitions. Information space is a vector space V of dimension $M \geqslant N$, where N is the number of terms, with each term corresponding to an independent axis. A document is given by a vector in the information space.

We consider V as a real vector space (with the usual inner product), and implement our intuitive idea of the closeness of concepts by insisting that the axes corresponding to two given terms have an angle given by the relation between the two terms (it is in fact the cosine of the angle). Two terms that occur in the same documents will tend to be highly related and hence will have a small angle, i.e., the corresponding axes will be nearly parallel.

Our quantitative measure of what the vector for a given document means is adjusted by this arrangement of angle between term axes^2. Two terms whose axes are nearly parallel say similar things about the documents they occur in. Similarly, two terms whose axes are nearly anti-parallel (we will use the term anti-parallel to indicate axes whose angle is near π, i.e., the cosine is near -1) say nearly opposite things about the documents they occur in.

Let T be the set of content stems. Let $B = \{ e_u \,/\, u \varepsilon T \}$be an orthonormal basis in V (something like the independent directions of meaning in the information space), and write each term using the basis B.

$$t \quad = \quad \mathop{}_{u\varepsilon T} a_{tu} e_u,$$

where the coefficients a_{tu} relate terms to distinctions in meanings (cf. componential semantics) (these values are not known now). By definition, $t \cdot u = R_{tu}$, so we have

$$R_{tu} \quad = \quad (\sum_{v\varepsilon T} a_{tv} e_v) \,.\, (\sum_{v\varepsilon T} \; a_u e_v) \quad = \quad \sum_{v\varepsilon T} a_{tv} a_{uv}, \text{ so that}$$

$$R \quad = \quad AA^t.$$

Therefore, the matrix R can only be interpreted in this way when it can be written in the form above. It is clear that R must be a real symmetric matrix. Also, by result 4.12.2 (p. 69 Marcus & Minc - A survey of matrix theory and matrix inequalities, PWS (1964)), R must be positive semidefinite.

Now if d is a document vector with weighted terms, we can write

$$d = \sum_t d_t t,$$

and for a query vector,

$$q = \sum_t q_t t.$$

Then we compute d.q as follows:

$$d.q = (\sum_t d_t t) . (\sum_u q_u u) = \sum_{t,u} d_t q_u t.u = \sum_{t,u} d_t q_u R_{tu} = dR^t q,$$

where d and q are considered as row vectors.

Similarly, the length $|d|^2 = d.d = dRd^t$ and the length $|q|^2 = q.q = qRq^t$. In order for the length to be nonzero for nonzero vectors, we must have R positive definite.

We write $\hat{d} = d/|d|$ for $d \neq 0$, $\hat{d} = 0$ for $d = 0$.

Now $\hat{d}.\hat{q} = \pm 1$ when d and q are parallel, and $\hat{d}.\hat{q}=0$ when d and q are perpendicular (unrelated), so we should retrieve the documents that have large values of this normalized inner product.

Another possibility is to take the length $|\hat{d}-\hat{q}|^2$, but we get the same information:

$$|\hat{d}-\hat{q}|^2 = |\hat{d}|^2 - 2\hat{d}.\hat{q} + |\hat{q}|^2 = 2(1-\hat{d}.\hat{q}).$$

A different normalization can be based on the Fisher discriminant, by taking our measure to be

$$\frac{|d-q|^2}{|d|^2+|q|^2} = \frac{|d|^2 + |q|^2 - 2\, d.q}{|d|^2 + |q|^2} = 1- \frac{2\, d.q}{|d|^2+|q|^2}$$

which has value 0 when d=q and its maximum value 1 when d.q=0 (for relations R with non-negative entries), or a little more than 1 when d and q are anti-parallel.

Since R is real symmetric, all the eigenvalues of R are real (by 4.7.18 p. 64 Marcus & Minc), and there is a real orthogonal matrix V such that $VRV^t = D$, for a real diagonal matrix D (orthogonal means $VV^t=V^tV=I$). Since rank A = rank AA^t always, by the Frobenius rank inequality (2.17.2 p. 27 in Marcus & Minc), we have rank D = rank R = rank A, so R is nonsingular if and only if the terms have linearly independent meanings (the terms linearly independent and statistically independent have no relation to each other). If we have insisted that R be positive definite, then this requirement is automatic.

The rows of V are a set of orthonormal eigenvectors of R, where row i is the eigenvector for the eigenvalue D_{ii}, i.e., a basis.

4.5. ANALYSIS OF A SQUARE-ROOT ALGORITHM.

The algorithm used by almost every floating-point square root subroutine is an implementation of the Newton-Raphson method. As is well-known this algorithm proceeds by iteration from a first guess which may be chosen in a number of ways. If the number for which we want the square root is A, and the first guess is x_o then the iteration formula is:

$$x_{i+1} = (x_i + A/x_i)/2,$$

for i=0,1..., until some standard of accuracy is met.

Consider first the theoretical case, where all computations can be made exact. Let $A > 0$, and let x_o be any positive number. We analyze the error in each element x_i. Suppose $S^2=A$, and define $E_i=x_i-S$ for each i(so E_i is the error at step i). Then

$$E_{i+1} = x_{i+1}-S = \frac{1}{2}(x_i^2 + A)/x_i - S$$

$$= \frac{1}{2}(x_i^2-2Sx_i + A)/x_i$$

$$= \frac{1}{2}\ ((x_i - S)^2 - S^2 + A)/x_i$$

$$= \frac{1}{2}\ E_i^{\ 2}/x_i$$

$$= E_i^{\ 2}/2x_i.$$

We note two facts about the error E_i. The first is that it is always positive for $i \geq 1$. (E_o may be negative, but the rest are positive). The second is that for $A > \frac{1}{4}$, each x_i for $i \geq 1$ is greater than $\frac{1}{2}$ (this fact follows from the first one), so that $E_{i+1} < E_i^{\ 2}$. In other words, if one has k bits of accuracy at step i (i.e., $\log_2(E_i) < -k$), than at step i+1, one has 2k bits of accuracy (i.e., $\log_2(E_{i+1}) < -2k$). It is this latter fact that makes the algorithm converge so quickly.

Our application requires a version of this algorithm using fixed-point arithmetic. It is a characteristic of the application that our results need not be perfectly accurate: if the square root is one or two off, it won't greatly matter. Furthermore, the main use of the algorithm will be for large numbers. We will describe and analyze an algorithm which makes an error of at most one, and none at all for large numbers.

The algorithm we use has three phases: we first check to see if the number is greater than 256. If so, we take an initial guess of b (see the equations to follow) and perform one iteration of a Newton-Raphson loop. If not, then we check to see if the number is at least

16. If so, we take an initial guess of b and perform one Newton-Raphson iteration. If the number is less than 16, we simply look up the answer in a table.

The three tables have size 128 words, 16 words and 16 bytes, and the entire program is less than 180 words.

Let A be between 2^{2k} and 2^{2k+1} (i.e., $2^{2k} \leq A < 2^{2k+1}$). For our application, we have the cases $k=4, \ell=7$, and also $k=2, \ell=4$. Write $A=2^{2k}\cdot q+r$, with $0 \leq r < 2^{2k}$, and look up the rounded square root b of $2^{2k}q$ (a table of size 2^{ℓ}). Thus

$$\begin{array}{lcccl}
b^2-b+1 & \leq & 2^{2k}q & < & b^2+b+1 \\
b^2-b & < & A & \leq & b^2+b+r < b^2+b+2^{2k} \leq (b^2+b)(1+1/q) \\
b\sqrt{1-\frac{1}{b}} & < & \sqrt{A} & < & b\sqrt{(1+1/b)(1+1/q)}
\end{array}$$

Our initial guess is $x_o=b$.

We have

$$\begin{array}{lclcl}
b^2+b & \geq & 2^{2k}q & > & b^2-b \\
4b^2+4b & \geq & 2^{2k+2}q & > & 4b^2-4b \\
(2b+1)^2 & > & 2^{2k+2}q & \geq & (2b-1)^2 \\
2b+1 & > & 2^{k+1}\sqrt{q} & \geq & 2b-1 \\
b+1 & > & 2^{k}\sqrt{q}+1/2 & \geq & b
\end{array}$$

We now need an estimate for the error of this initial guess. For this purpose, we recall that for $x^2 < 1$,

$$(1 \pm x)^{\frac{1}{2}} = 1 \pm \frac{1}{2}x - \frac{1}{2.4}x^2 \pm \frac{1.3}{2.4.6}x^3 - \frac{1.3.5}{2.4.6.8}x^4 \pm \dots,$$

so that $\sqrt{1+1/b} < 1 + \frac{1}{2b}$ and

$$b\sqrt{1+1/b} < b + 1/2.$$

Also, $\sqrt{1-1/b} = 1 - \frac{1}{2b} - \frac{1}{2.4b^2} - \frac{1.3}{2.4.6b^3} - \dots,$

$$= 1 - \frac{1}{2b} - \frac{1}{8b^2}\left(1 + \frac{1}{2b} + \frac{5}{2.8b^2} + \frac{5.7}{2.8.10b^3} + \dots\right)$$

$$> 1 - \frac{1}{2b} - \frac{1}{8b^2}\left(\frac{1}{2} + \frac{1}{2}\left(1 + \frac{1}{b} + \frac{1}{b^2} + \dots\right)\right)$$

$$= 1 - \frac{1}{2b} - \frac{1}{16b^2}\left(1 + 1/(1-\frac{1}{b})\right),$$

so

$$b\sqrt{1-1/b} > b - 1/2 - (1 + b/(b-1)/16b$$

$$= b - 1/2 - 3(2 + 1/(b-1))/16b$$

$$> b - 1/2 - \frac{3}{16b}$$ since $b \geq 2 \Rightarrow \frac{b}{b-1} \leq 2.$

Thus

$$b - 1/2 - 3/16b < \sqrt{A} < (b+1/2)(1+1/q)^{\frac{1}{2}}$$

$$(b+1/2)\ (1+ 1/2q)$$

$$-\ 1/2 - \frac{3}{16b} < E_o < \frac{1}{2} + \frac{1}{4q} + \frac{b}{2q}$$

$$-\ \frac{9}{16} < E_o < 1/2 + \frac{2b+1}{4q}$$

Now: $k \geq 2$ gives $b+1 > 2^2 \sqrt{q} + .5 \geq 4.5$, so $b > 3.5$ and $b \geq 4$.

We want $E_1 < 1$.

Since $E_1 = E_o^2/2b$, we can check that $E_1 < 1$ for negative values of E_o easily, since then $|E_o| < 9/16$. It follows that

$E_o^2 < 81/256$, so $b \geq 4$ implies

$E_1 < 81/2048$

Therefore, the only problem that occurs will be at the other side, i.e., for positive values of E_o.

Since $E_1 = E_o^2/2b < 1$ iff

$E_o^2 < 2b$ iff

$E_o < \sqrt{2b}$ and $b \geq 3$, it suffices for $k=2$ to show that $E_o < \sqrt{8}$, and for $k \geq 4$, we have

$$b+1 > 2^4\sqrt{q} + 1/2 \geq 16.5 \text{ , so}$$

$$b > 15.5\text{, so}$$

$$b \geq 16.$$

It suffices for $\geq k$ 4 to show that $E_o < \sqrt{32}$. In general, for a given $k \geq 1$, we have $b \geq 2^k$ so that it suffices to show that

$$E_o < \sqrt{2^{k+1}}.$$

We know that $E_o < 1/2 + \frac{2b+1}{4q}$, and that $b \leq 2^k\sqrt{q+1/2}$, so

$$E_o < 1/2 + (2^k \sqrt{q} + 1)/2q\text{, or}$$

$$E_o < (q + 2^k\sqrt{q} + 1)/\ 2q.$$

Since the number on the right here is decreasing as q increases, it suffices to check the small values.

For $k \geq 1$, we have that for $q=4$, we are comparing $(5+2^{k+1})/8$ with $\sqrt{2^{k+1}}$,

$$5 + 2^{k+1} \text{ with } 2^3 \ . \ 2^{(k+1)/2} = 2^{(k+7)/2}.$$

For k = 2,4,6 these values are

$5+8=13$ with $2^{9/2} > 2^4=16$

$5+32=37$ with $2^{11/2} > (2025)^{\frac{1}{2}} = 45$

$5+128=133$ with $2^{13/2} = (8192)^{\frac{1}{2}} < 100$, so that for our case, the error is within the prescribed bounds for $q \geq 4$.

Finally, for $1 \leq q \leq 3$, we check the earlier equation.

$$b > 2^k \sqrt{q} - 1/2$$

$$E_o < (q + 2^k \sqrt{q+1})/2q$$

and we want $e_o < \sqrt{2b}$

k	q	$2^k \sqrt{q}$	b	numerator	bound	$\sqrt{2b}$
2	1	4	4	6	3	2.828
2	2	5.656	6	8.656	2.164	6.928
2	3	6.928	7	10.928	1.821	> 3
4	1	16	6	18	9	< 6
4	2	22.626	23	25.626	6.406	6.782
4	3	27.712	28	31.712	5.285	> 6

So the only remaining difficulty is with q=1, where our initial guess (namely b) is not close enough to guarantee an error of less than one. The relevant arguments are (for k=2) numbers from 16 to 31, where we guess 4 and (for k=4) numbers from 256 to 511, where we guess 16.

The easiest way to fix this problem is to change the initial guesses in these cases. If we choose to guess 5 for numbers between 16 and 31, then the initial error E_0 is at most 1, so the error E_1 is less than 1. Since numbers between 256 and 511 have square roots ranging from 16 almost up to $23 = \sqrt{529}$, an initial guess of 19 gives an error E_0 of less than 4. Then the error E_1 is less than 1.

We point out here that the above error analysis is not applicable directly, because the calculations in the Newton-Raphson iteration are also done in fixed point. It turns out that the proper analysis isn't quite so easy, and also that the results are better: with no change in the initial guess, we get the correct answer (error no more than 1), though not necessarily the closest answer (this would require an error of less than 1/2).

It is to be noted that since the error E_1 is always positive, and since fixed point calculations are all truncated (instead of rounded), we could expect to get the truncated square root almost always. A more detailed analysis of the algorithm will support this proposition.

4.6. QUERY FORMATION

Initial system designs include generation of a single query vector for each user request. There is no reason, however, that a system cannot generate several query vectors, which could be used consecutively.

After completing a search with the first query vector, documents with correlations below some minimum threshold could be eliminated, and succeeding query vectors could be applied to get successively smaller sets of documents.

The thresholds and the order in which to take multiple query vectors will have to be determined by experiment because of the lack of previous work in this area. It may be that the extra precision gained by multiple query vectors is insufficient to justify the added overhead of maintaining lists of documents.

The following experiments are planned:

- Multiple query vectors. This will consist of observing the effect of splitting a query vector into varying numbers of new query vectors. Final document-query correlations will be the product of the correlations between the document and each of the multiple queries, with correlations below some threshold being set to zero.

- Recognizing dependency. Syntactic dependency relations such as subject-verb and adjective-noun are not recognized by traditional associative analysis systems (such as SMART) in constructing query vectors. It is possible to approximate the effect of syntactic dependency by splitting a query vector into levels, where dependent stems occur at levels below the stems on which they depend. This in general will require a query in the form of a tree, but advantages may be gained from splitting a query simply into a dominant and a dependent query vector. Experiments for testing this approach are under way.

4.7. OPTIONAL ADDITIONS TO THE BASIC SYSTEM

In addition to areas of experimentation discussed under the preceding headings, other experiments will consider fairly direct additions to the I&W system, independent of its basic structure. These will include:

- Immediate keyword retrieval of documents on their raw stems (without statistical processing). This will bypass the start-up process in order to pick up messages as soon as they arrive, and thus will add near-real-time facilities to the system. It also may earmark certain messages for additional analysis.

- Background processing. This option would allow a user to specify queries to be run immediately after startup, without requiring the user to enter them each day, or even to be present at a console. It also would provide input for the selection of content stems in startup.

- User background information. This option would involve the system monitoring the content of the user's queries to determine his special interests. Such information would be added automatically to subsequent queries, providing an "automatic context." This would also be used to help select content stems.

- o Topicalization. Extra weight could be assigned to stems at pre-designated syntactic positions within a message. This may give a more accurate picture of what a text is about than the usual way of counting.

- o Complemented or negated features. As an extra convenience, a user ought to be able to say in his query what he does not want.

- o User-specified weightings. It should be possible to specify the weightings to be given to each word in a query through some easy and obvious convention.

- o Query language keywords. This option would allow a user to include command information within a query. Such an approach would permit a sophisticated user to avoid lengthy question-and-answer sessions with an interactive system by proceeding directly to required system actions.

- o Headers. There should be a separate component for matching messages against their headers without applying associative techniques.

- o Alternative measures. The present statistical filter is not necessarily the best way of deciding whether a stem should be a content stem. It should be possible to develop a hybrid approach with various stop-word lists (i.e., lists of words to be excluded from the thesaurus) and various alternative weighting schemes.

All the preceding experiments will be carried out by implementing several mini-versions of the system in the high-level language PASCAL. The choice of language here is for the purpose of easing programming and instrumentation and does not reflect a commitment to the PASCAL language for implementation of the actual prototype system.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL
1776-1976