DOCUMENT RESUME

ED 048 912 | LI 002 721

TITLE	Automatic Dictionary Construction; Part II of Scientific Report No. ISR-18, Information Storage and Retrieval...
INSTITUTION	Cornell Univ., Ithaca, N.Y. Dept. of Computer Science.
SPONS AGENCY	National Library of Medicine (DHEW), Bethesda, Md.; National Science Foundation, Washington, D.C.
REPORT NO	ISR-18[Part II]
PUB DATE	Oct 70
NOTE	124p.; Part of LI 002 719
EDRS PRICE	EDRS Price MF-$0.65 HC-$6.58
DESCRIPTORS	Automation, *Dictionaries, *Information Retrieval, Lexicography, *Lexicology, *Search Strategies, *Thesauri, Vocabulary, Word Lists
IDENTIFIERS	On Line Retrieval Systems, *Saltons Magical Automatic Retriever of Texts, SMART

ABSTRACT

Part Two of the eighteenth report on Salton's Magical Automatic Retriever of Texts (SMART) project is composed of three papers: The first: "The Effect of Common Words and Synonyms on Retrieval Performance" by D. Bergmark discloses that removal of common words from the query and document vectors significantly increases precision and that synonyms were more effective for recall than common words. Paper two: "Negative Dictionaries" by K. Bonwich and J. Aste-Tonsmann discusses a rationale for constructing negative dictionaries and examines the retrieval results of experimentally produced dictionaries. The third paper: "Experiments in Automatic Thesaurus Construction for Information Retrieval" by G. Salton describes several new methods for automatic, or semi-automatic, dictionary construction, including procedures for the automatic identification of common words, and novel automatic grouping methods. The resulting dictionaries are evaluated in an information retrieval environment. (For the entire SMART project report see LI 002 719, for Part One see LI 002 720 and for Parts 3-5 see LI 002 722 through LI 002 724.) (NH)

Department of Computer Science

Cornell University

Ithaca, New York 14850

Automatic Dictionary Construction Part II

of

Scientific Report No. ISR-18

INFORMATION STORAGE AND RETRIEVAL

to

The National Science Foundation

and to

The National Library of Medicine

Reports on Analysis, Dictionary Construction, User Feedback, Clustering, and On-Line Retrieval

Ithaca, New York

October 1970

Gerard Salton

Project Director

2

SMART Project Staff

Robert Crawford
Barbara Galaska
Eileen Gudat
Marcia Kerchner
Ellen Lundell
Robert Peck
Jacob Razon
Gerard Salton
Donna Williamson
Robert Williamson
Steven Worona
Joel Zumoff

ERIC User Please Note:

This Table of Contents outlines all 5 parts of Information Storage and Retrieval (ISR-18), which is available in its entirety as LI 002 719. Only the papers from Part Two are reproduced here as LI 002 721. See LI 002 720 for Part One and LI 002 722 thru LI 002 724 for Parts 3 - 5.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

PART TWO

AUTOMATIC DICTIONARY CONSTRUCTION

TABLE OF CONTENTS (continued)

Page

V. continued

"The Effect of Common Words and Synonyms on Retrieval Performance"

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

Page

PART THREE

USER FEEDBACK PROCEDURES

Available as LI 002 722

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

Page

PART FOUR

CLUSTERING METHODS

Available as
LI 002 723

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

ERIC User Please Note:

This summary discusses all 5 parts of Information Storage and Retrieval (ISR-18), which is available in its entirety as LI 002 719. Only the papers from Part Two are reproduced here as LI 002 721. See LI 002 720 for Part One and LI 002 722 thru LI 002 724 for Parts 3 - 5.

Summary

The present report is the eighteenth in a series describing research in automatic information storage and retrieval conducted by the Department of Computer Science at Cornell University. The report covering work carried out by the SMART project for approximately one year (summer 1969 to summer 1970) is separated into five parts: automatic content analysis (Sections I to IV), automatic dictionary construction (Sections V to VII), user feedback procedures (Sections VIII to XI), document and query clustering methods (Sections XII and XIII), and SMART systems design for on-line operations (Sections XIV and XV).

Most recipients of SMART project reports will experience a gap in the series of scientific reports received to date. Report ISR-17, consisting of a master's thesis by Thomas Brauen entitled "Document Vector Modification in On-line Information Retrieval Systems" was prepared for limited distribution during the fall of 1969. Report ISR-17 is available from the National Technical Information Service in Springfield, Virginia 22151, under order number PB 186-135.

The SMART system continues to operate in a batch processing mode on the IBM 360 model 65 system at Cornell University. The standard processing mode is eventually to be replaced by an on-line system using time-shared console devices for input and output. The overall design for such an on-line version of SMART has been completed, and is described in Section XIV of the present report. While awaiting the time-sharing implementation of the system, new retrieval experiments have been performed using larger document collections within the existing system. Attempts to compare the performance

of several collections of different sizes must take into account the collection "generality". A study of this problem is made in Section II of the present report. Of special interest may also be the new procedures for the automatic recognition of "common" words in English texts (Section VI), and the automatic construction of thesauruses and dictionaries for use in an automatic language analysis system (Section VII). Finally, a new inexpensive method of document classification and term grouping is described and evaluated in Section XII of the present report.

Sections I to IV cover experiments in automatic content analysis and automatic indexing. Section I by S. F. Weiss contains the results of experiments, using statistical and syntactic procedures for the automatic recognition of phrases in written texts. It is shown once again that because of the relative heterogeneity of most document collections, and the sparseness of the document space, phrases are not normally needed for content identification.

In Section II by G. Salton, the "generality" problem is examined which arises when two or more distinct collections are compared in a retrieval environment. It is shown that proportionately fewer nonrelevant items tend to be retrieved when larger collections (of low generality) are used, than when small, high generality collections serve for evaluation purposes. The systems viewpoint thus normally favors the larger, low generality output, whereas the user viewpoint prefers the performance of the smaller collection.

The effectiveness of bibliographic citations for content analysis purposes is examined in Section III by G. Salton. It is shown that in some situations when the citation space is reasonably dense, the use of citations attached to documents is even more effective than the use of standard keywords or descriptors. In any case, citations should be added to the normal descriptors whenever they happen to be available.

In the last section of Part 1, certain template analysis methods are applied to the automatic resolution of ambiguous constructions (Section IV by S. F. Weiss). It is shown that a set of contextual rules can be constructed by a semi-automatic learning process, which will eventually lead to an automatic recognition of over ninety percent of the existing textual ambiguities.

Part 2, consisting of Sections V, VI and VII covers procedures for the automatic construction of dictionaries and thesauruses useful in text analysis systems. In Section V by D. Bergmark it is shown that word stem methods using large common word lists are more effective in an information retrieval environment that some manually constructed thesauruses, even though the latter also include synonym recognition facilities.

A new model for the automatic determination of "common" words (which are not to be used for content identification) is proposed and evaluated in Section VI by K. Bonwit and J. Aste-Tonsmann. The resulting process can be incorporated into fully automatic dictionary construction systems. The complete thesaurus construction problem is reviewed in Section VII by G. Salton, and the effectiveness of a variety of automatic dictionaries is evaluated.

Part 3, consisting of Sections VIII through XI, deals with a number of refinements of the normal relevance feedback process which has been examined in a number of previous reports in this series. In Section VIII by T. P. Baker, a query splitting process is evaluated in which input queries are split into two or more parts during feedback whenever the relevant documents identified by the user are separated by one or more non-relevant ones.

The effectiveness of relevance feedback techniques in an environment of variable generality is examined in Section IX by B. Capps and M. Yin. It is shown that some of the feedback techniques are equally applicable to collections of small and large generality. Techniques of negative feedback (when no relevant items are identified by the users, but only nonrelevant ones) are considered in Section X by M. Kerchner. It is shown that a number of selective negative techniques, in which only certain specific concepts are actually modified during the feedback process, bring good improvements in retrieval effectiveness over the standard nonselective methods.

Finally, a new feedback methodology in which a number of documents jointly identified as relevant to earlier queries are used as a set for relevance feedback purposes is proposed and evaluated in Section XI by L. Paavola.

Two new clustering techniques are examined in Part 3 of this report, consisting of Sections XII and XIII. A controlled, inexpensive, single-pass clustering algorithm is described and evaluated in Section XII by D. B. Johnson and J. M. Lafuente. In this clustering method, each document is examined only once, and the procedure is shown to be equivalent in certain circumstances to other more demanding clustering procedures.

The query clustering process, in which query groups are used to define the information search strategy is studied in Section XIII by S. Worona. A variety of parameter values is evaluated in a retrieval environment to be used for cluster generation, centroid definition, and final search strategy.

The last part, number five, consisting of Sections XIV and XV, covers the design of on-line information retrieval systems. A new SMART system design for on-line use is proposed in Section XIV by D. and R. Williamson, based on the concepts of pseudo-batching and the interaction of a cycling program with a console monitor. The user interface and conversational facilities are also described.

A template analysis technique is used in Section XV by S. F. Weiss for the implementation of conversational retrieval systems used in a time-sharing environment. The effectiveness of the method is discussed, as well as its implementation in a retrieval situation.

Additional automatic content analysis and search procedures used with the SMART system are described in several previous reports in this series, including notably reports ISR-11 to ISR-16 published between 1966 and 1969. These reports are all available from the National Technical Information Service in Springfield, Virginia.

G. Salton

V. The Effect of Common Words and Synonyms on Retrieval Performance

D. Bergmark

Abstract

The effect of removing common words from document and query vectors is investigated, using the Cran-200 collection. The method used is comparison of a standard stem dictionary and a thesaurus with a new dictionary formed by adding an extensive common word list to the standard stem dictionary. It is found that removal of common words from the query and document vectors significantly increases precision. Query and document vectors without either common words or synonyms yield the highest precision results but inferior recall rssults. Synonyms are found to be more effective for recall than common words.

1. Introduction

A thesaurus results in about 10% better retrieval than a standard stem dictionary, according to results in previous studies [2]. This fact leads to the question of why the thesaurus performs better: is it because it groups terms into synonym classes, or is it because the thesaurus includes a large common word list. If both contribute to the superiority of the thesaurus, then it is desirable to determine what proportion of this improvement is due to each factor. Taking common words out of a thesaurus could consume little time compared to that required for grouping concepts into synonym classes if an appropriate means of automatically generating the common word list were found. Therefore, if a large amount of improvement of a thesaurus over the stem dictionary is due to removing common words and putting them in a separate list, then it would be advantageous to devote work to methods of isolating the insignificant words.

The subject of this paper, then, is a comparison of the search results of a standard stem dictionary, a thesaurus, and a standard stem dictionary with an extensive common word list. The results of this study indicate that a large amount of the difference in retrieval performance between thesaurus and standard stem dictionaries is due to the removal of common words into a separate list. Surprisingly, the effect of synonyms and of common words are similar; both encourage higher recall but both degrade precision.

2. Experiment Outline

A) The Experimental Data Base

With limited resources, it is fairly important to choose carefully the collection to be studied. First, the collection must be small enough to be manageable within the resources available, yet large enough to give significant results. The collection also has to have both a thesaurus and a word stem dictionary available.

The Cran-200 collection seems to satisfy these criteria and is chosen as the basis for the study. This collection has 200 documents and 42 queries, and the text is available on tape for lookup with a new dictionary.

B) Creation of the Significant Stem Dictionary

Investigating the retrieval effectiveness of an extensive common word list together with a standard stem dictionary requires, per force, the generation of a new dictionary. Specifically, the new dictionary desired is one which has the same stems as the standard stem dictionary but with many more words marked as common.

The most readily available common word list for the Cran-200 collection is contained in the Cran-200 thesaurus. In fact, the thesaurus is essentially the same dictionary as the standard stem dictionary except that many more words are flagged as common, and synonyms are grouped into concept classes by assignment of the same concept number to all word stems synonymous with each other. Furthermore, since the same word may occur in more than one concept class, one term may have more than one concept number assigned to it.

Thus more "significance" decisions are made in constructing a thesaurus than in constructing a standard stem dictionary, both in removing common and in removing infrequently used words from the dictionary list. Hence if a thesaurus is turned back into a standard stem dictionary, the result is a standard stem dictionary with a large common word list. Therefore, rather than going through the standard stem dictionary and marking additional words as common, the strategy followed in this experiment is to go through the thesaurus and renumber the words so that the common words are still flagged as common, but the stems are separated so that no two stems have the same concept number and each stem has only one concept number. This method is efficient since no word-matching need be done to determine which are common words and which are not.

Punching the Cran-200 thesaurus, CRTHES, from Tape 9 onto cards yields approximately 3380 cards with one thesaurus term per card along with its concept class(es). These cards are then used as input to a 360/20 RPG program which punches a duplicate deck in which each thesaurus term is assigned a unique concept number, with numbering starting at 1 for the significant terms and at 32001 for common terms. This results in 2940 significant, distinct words and 741 distinct common words.

That the resulting dictionary (henceforth referred to as the "significant stem dictionary") is the one desired can be seen from Appendix I, which lists some typical query vectors using each of the three dictionaries. It can be seen that the significant and standard stem queries are sufficiently similar except for the inclusion of common words in the standard stem queries.* The significant stem dictionary has approximately twice as many words marked as common than does the standard stem dictionary. In addition, the significant stem dictionary has about 65% as many significant concepts as the standard, and many of the remainder are actually common and so were never included, or were deleted from, the thesaurus. The new dictionary thus has the same word significance decisions (i.e., the same common word list) as the thesaurus, but the same grouping decisions (i.e., none) as the word stem dictionary.

C) Generation of New Query and Document Vectors

With the creation of the new dictionary, it is necessary to reassign vectors for the queries and documents of the Cran-200 collection in preparation for search runs. To accomplish this task the LOOKUP program, written in PL/I, is used. This program reads in a dictionary, a suffix list, and the query or document texts; it then generates concept vectors for the texts using the standard suffixing rules. It is run once for the queries and once for the documents.

Some decision has to be made concerning the suffix list; ideally it should be as close as possible to that used for creating the original thesaurus and standard stem vectors for the Cran-200 collection. The suffix list used in this study contains approximately 195 terms, and the resulting vectors indicate that it is quite similar to the one used to generate thesaurus and standard stem vectors.

---

*There was some concern in the early stages of this work that the thesaurus contains many full words rather than stems. Although there are full words in the thesaurus which are only stems in the stem dictionary, the reverse is also true. In any case, analysis of individual queries shows that these discrepancies have no significant effect on what is retrieved.

As far as the Cran-200 text is concerned, it has to be picked out from the Cran-1400 collection. A slight modification of the LOOKUP program does this by allowing the user to specify which of the Cran-1400 query and document texts are to be processed. One Cran-200 text (Text 995) is not on the Cran-1400 tape but is fortunately not relevant to any of the Cran-200 queries; it is not believed that the missing document perturbs results very much.

The average length of the resulting significant stem queries is 6.14 words as opposed to the standard stem queries with 8.26 words and the thesaurus queries with 6.98 words. The size of the document vectors varies proportionally with the length of the queries, except that the thesaurus document vectors are in general slightly shorter than the significant stem document vectors.

Why there are more words in the thesaurus queries than in the significant stem queries is somewhat unclear. As can be seen from the queries listed in Appendix I the additional words in the thesaurus queries are common ones; these words have been removed from the thesaurus, probably because they were judged to be common, and thus do not appear in the significant stem queries. On the other hand, some thesaurus queries have fewer significant terms than the significant stem queries; this is because if two words are synonymous, their concept number appears only once in the thesaurus query with a heavier weight.

D) Document Analysis — Search and Average Runs

In order that the evaluation of all three dictionaries is on a consistent basis, search runs must be done using vectors generated with all three dictionaries. Relevancy judgments must be added to the significant stem query vectors obtained by LOOKUP so that the same relevancy judgments are used for each of the three sets of queries. A fairly simple search without complex parameters is performed so that unnecessary complications in analysis do not arise. A full search lists the top thirty documents, and then a positive feedback search using the top five documents is done to make sure that removing common words and synonyms does not have an unforseen effect on feedback.

The results of the three searches, thesaurus, significant stem and standard stem, are compared by analysis of overall measures as well as in-depth analysis of individual queries to see to what extent not having synonyms hurt or help the retrieval process. Similarly, in-depth analysis is required to see what effect common words, or lack of them, have on retrieval.

To aid the analysis, the standard averages are obtained as well as the recall-level and document-level recall-precision graphs. The three full searches are compared with each other, and the three feedback runs are compared with each other. Results are verified using the standard significance tests.

In addition, some statistics are calculated by hand to determine retrieval effectiveness. Specifically, it is felt that the default rank recall measure provided in the SMART averaging routines is not quite suited to the analysis being done here. When some of the relevant documents do not have any correlation with the query, the averages have to be based on extrapolation; in the standard SMART run, the rank recall is calculated assuming that the relevant documents with no correlation appear at the bottom of the list (i.e., rank 200, 199, 198,...). Since this project is directed toward seeing what effect common words have on precision as well as recall, it seems better to take into account the number of documents, relevant and non-relevant, which correlate with the query in the first place. That is, it seems that if one is testing precision, and if two queries each retrieve six out of nine relevant documents, but one of them recovers thirty more non-relevant documents than the other before going on to a zero correlation, it should be judged less precise than the other. Thus in the graphs derived by hand, rank recall is extrapolated on the basis of CORR.RANK+1, CORR.RANK+2, etc. for the relevant documents which have zero correlation with the query.

All in-depth analysis is performed on the full search results rather than on feedback results because the project is more concerned with determining the effect of dictionaries rather than the effect of feedback on retrieval. The recall-precision graphs for the three feedback runs are, however, included in Appendix II.

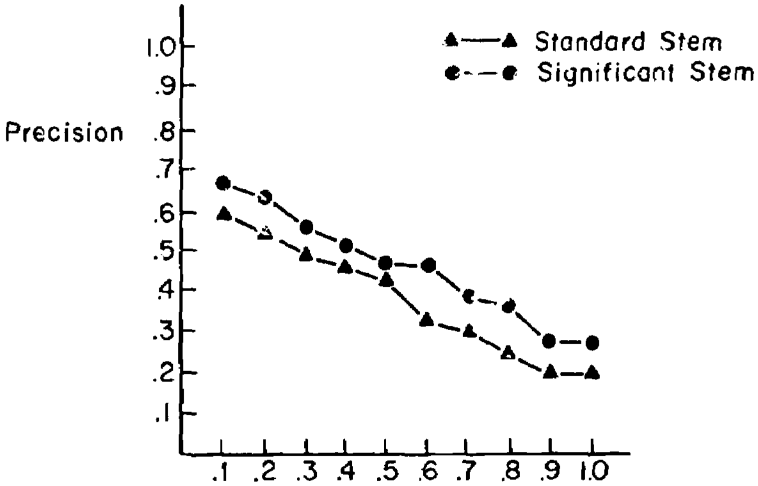
3. Retrieval Performance Results

A) Significant vs. Standard Stem Dictionary

The results of this experiment show that, as expected, use of a large common word list improves the retrieval performance of a standard stem dictionary. It can be seem from Graphs 1 and 2, which show the recall and precision averages for two full searches, one using the standard stem dictionary and the other using the significant stem dictionary, that the significant stem dictionary results in greater precision at all recall and document levels.

Furthermore, global statistics for these runs bear out the same conclusion, that the significant stem performs better than the standard stem:

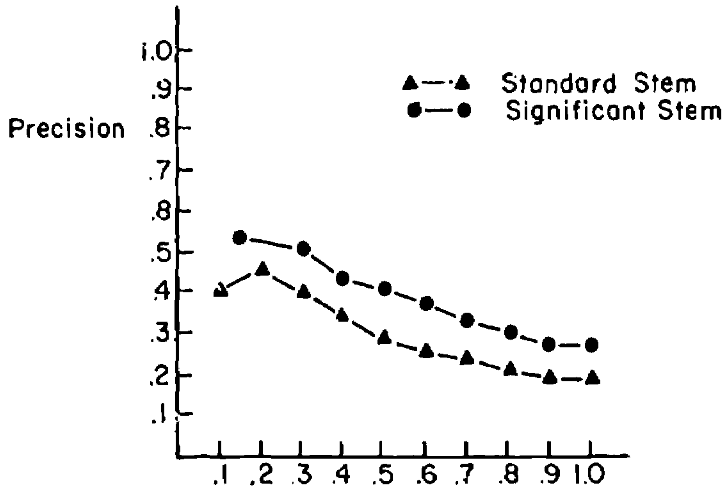
	Standard Stem	Significant Stem
Rank Recall	.2424	.3331
Log Precision	.4202	.5053



Recall

Significant vs. Standard Stem
Recall-Level Averages
Full Search

Graph 1



Recall

Significant vs. Standard Stem
Document-Level Averages
Full Search

Graph 2

The above statistics are significant according to all the usual significance tests.

It is interesting to note that the difference between the significant and standard stem curves remains fairly constant despite the recall or document level. This indicates that the significant stem performs roughly the same retrieval as the standard stem, only more precisely. In other words, including common terms in the document and query vectors seems to uniformly degrade precision performance.

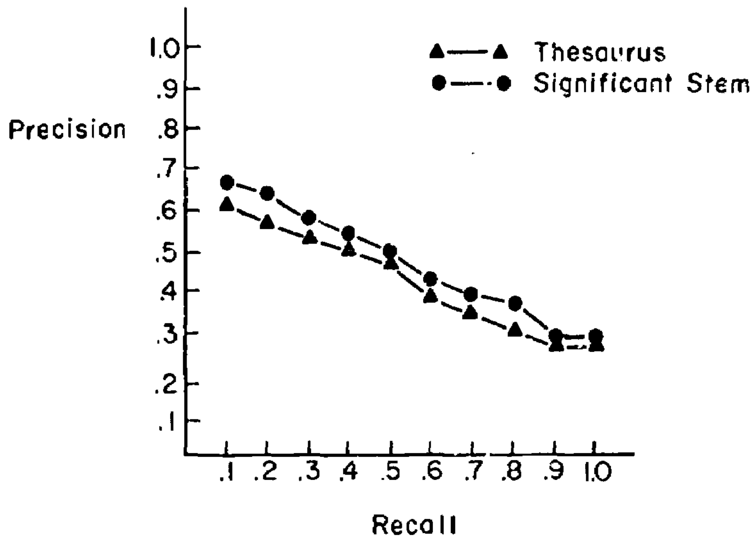
B) Significant Stem vs. Thesaurus

It was originally expected ttat using a standard stem dictionary with a large common word list would result in search performance better than the standard stem but not as good as the thesaurus. From the recall-precisi : Graphs 3 and 4 it can be seen that contrary to these expectations the significant stem performs just as well as the thesaurus, if not better.

The similarity of the significant stem and thesaurus curves is confirmed by global statistics, which while extremely close give a slight edge to the significant stem dictionary:

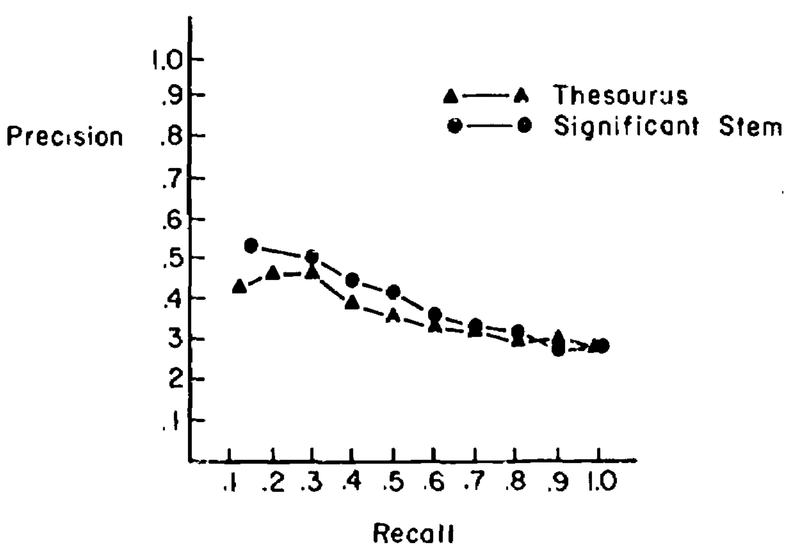
	Significant Stem	Thesaurus
Rank Recall	.3331	.3222
Log Precision	.5053	.4880

Here the difference between the two curves is not the same. The significant stem performs better than the thesaurus at the low end of the curve, but loses this edge as recall increases. One may conclude that the standard stem queries find only the first few relevant documents faster than



Significant Stem vs. Thesaurus
Recall-Level Averages
Full Search

Graph 3



Significant Stem vs. Thesaurus
Document-Level Averages
Full Search

Graph 4

the thesaurus.

C) Standard Stem vs. Thesaurus

In general a thesaurus results in better retrieval performance than a standard stem dictionary, and this experiment has roughly the same appearance. Recall-Precision Graphs 5 and 6 indicate the superiority of the thesaurus over the standard stem at all recall and document levels, with the superiority most marked at high recall levels. That the thesaurus, with its common word list and synonyms, is better than the standard stem but is approximately equal to the significant stem, with only a common word list, indicates that much of the improvement of the thesaurus over the standard stem is due to the common word list. Furthermore, comparison of these three sets of recall-precision plots seems to indicate that at the low recall end synonyms actually degrade precision, acting as common words do.

D) Recall Results

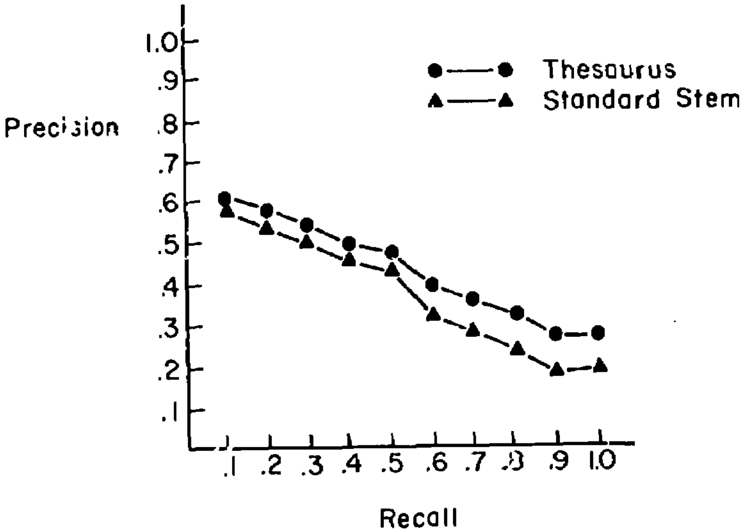
The difficulty with the significant stem dictionary, however, can be detected in the normalized global statistics (Figure 1).

	Standard Stem	Significant Stem	Thesaurus
Norm Recall	.8489	.8330	.8732
Norm Precision	.6615	.6918	.6924

Normal Recall and Precision for Full Search, All Dictionaries

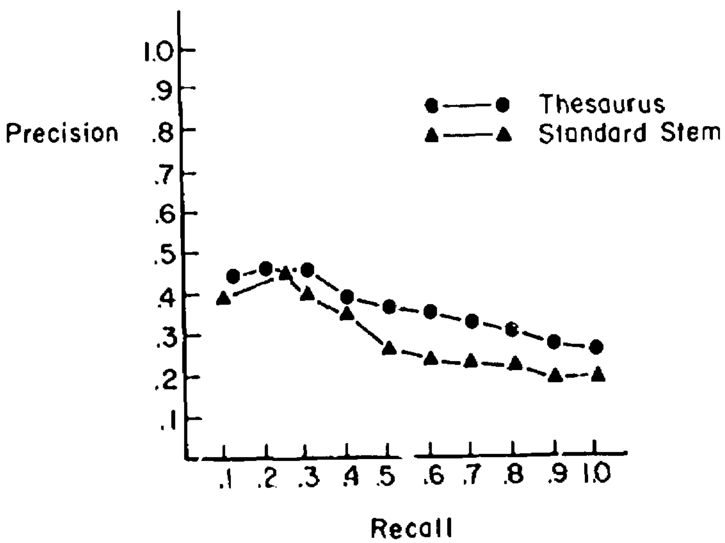
Figure 1

These global statistics are much closer than the Rank Recall and Log Precision and indeed, the first favors the standard stem dictionary over the significant stem although neither are significantly different according



Thesaurus vs. Standard Stem
Recall-Level Averages
Full Search

Graph 5



Thesaurus vs Standard Stem
Document - Level Averages
Full Search

Graph 6

to the t-test. The problem displayed here is that the significant stem ultimately results in lower recall than does the standard stem; more queries have rank and precision measures based on extrapolation in the first case than in the second.

To be specific, 14 of the 42 queries using the significant stem dictionary do not have a 1.00 recall ceiling during the full search, while only nine of the standard stem and six of the thesaurus do not. The average recall ceiling for the significant stem is 0.8853 while the average recall ceiling for the standard stem is 0.9390 and 0.9565 for the thesaurus. After feedback, however, the difference narrows somewhat, going to 0.9504 for the significant stem dictionary and 0.9841 for the standard stem dictionary (the thesaurus at 0.9814 after feedback is not quite as good as the standard stem dictionary).

It is reasonable that the recall ceiling is higher for the standard stem than for the significant stem, since the average query length for the latter is greater than that for the former. Thus chances for a significant stem query not correlating at all with documents relevant to it are greater than those for a standard stem query. Similarly synonyms improve the chances for the thesaurus query's matching at least one relevant document.

To measure this recall difference in another way, Figure 2 displays a recall measure used by Keene [2] based on the average rank of the last relevant document retrieved. Figure 2 is based on the full search results.

The method 1 averages, which measure ultimate recall ability, shows that the thesaurus is superior in this respect, while the significant stem dictionary has the poorest recall. The method 2 averages, however, which are more a measure of precision in that they also include a measure of how many non-relevant documents are retrieved before correlation goes to zero,

Dictionary	Method 1	Method 2
Standard Stem	83.33	60.29
Significant Stem	87.64	46.45
Thesaurus	73.24	57.57

Method 1: Unrecovered relevant documents assigned ranks of 200, 199, etc.

Method 2: Unrecovered relevant documents assigned ranks of CORR.RANK+1, CORR.RANK+2, etc. where CORR.RANK is the rank of the documents with the lowest correlation with the query greater than 0.

Average Rank of the Last Relevant Document

Figure 2

put the significant stem at the top of the list. Thus these averages reinforce the previous hypothesis that if the user wants to recover every last relevant document, he should use the thesaurus, and if instead he is interested in minimizing the number of non-relevant retrieved, he should use the significant stem dictionary.

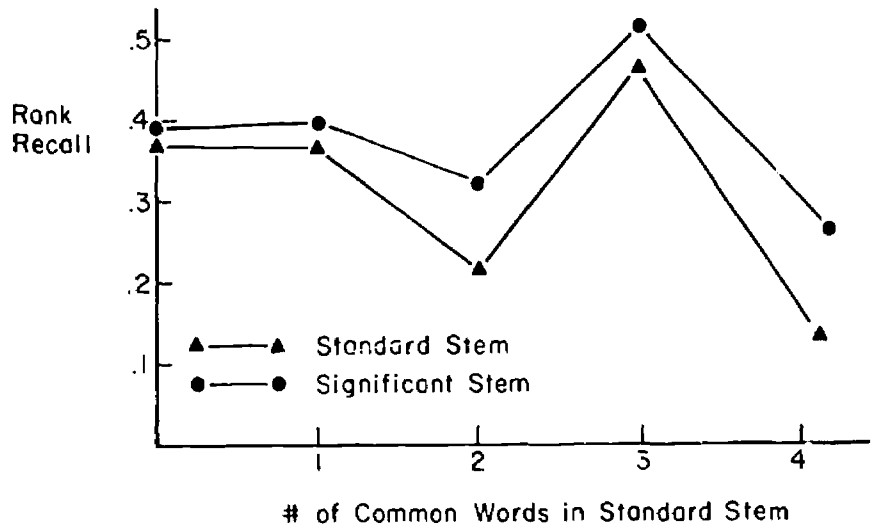
E) Effect of "Query Wordiness" on Search Performance

While it seems clear that significant stem results in an overall increase in precision over standard stem queries, it seems likely that the "wordiness" of a query, or the number of common words included in the standard stem query not included in the signifi[illegible] query, should have some effect on retrieval. That is, the more v[illegible] standard stem query is, the more non-relevant documents should be r[illegible] before all the relevant ones. Graph 7 shows the rank recall averag[illegible] standard and significant stems, over all 42 queries, at various lev[illegible] rdiness".

It is not really clear that retrieval d[illegible] faster as more and more common words are added to the query. [illegible] of possible explanations for this are 1) all the common words toge[illegible] etrieve the same documents, since the common words in a given qu[illegible] be "related", or 2) of the common words added, only one or two of [illegible] responsible for retrieving garbage. (The latter theory seems t[illegible] firmed by study of individual queries.) The left part of the gra[illegible] course identical for both dictionaries since at that point the queri[illegible] practically identical.

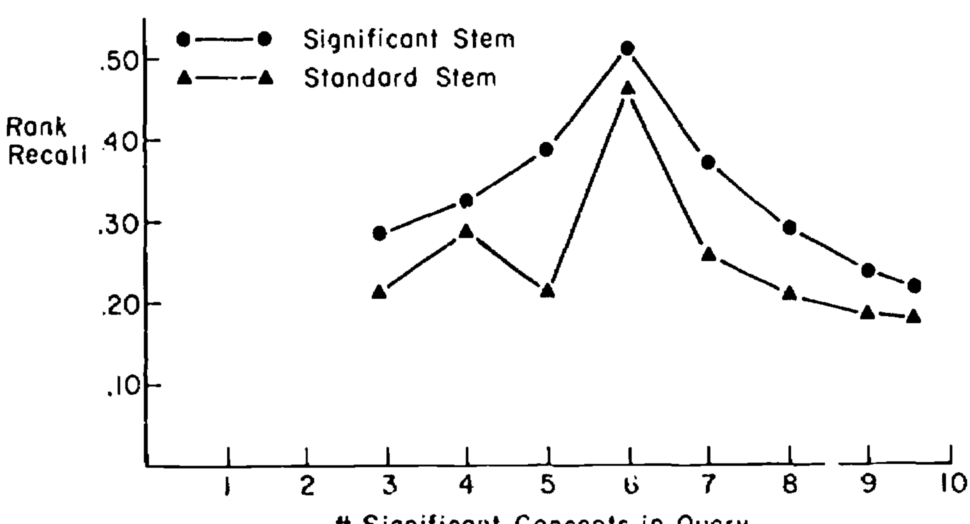
F) Effect of Query Length on Search Per[illegible]

It also seems likely that the differen[illegible] performance would vary depending on the number of significant concept[illegible] query. For example, if the significant stem query is very explicit, c[illegible]ing many significant



Rank Recall vs. Wordiness

Graph 7



Length of Query vs. Rank Recall

Graph 8

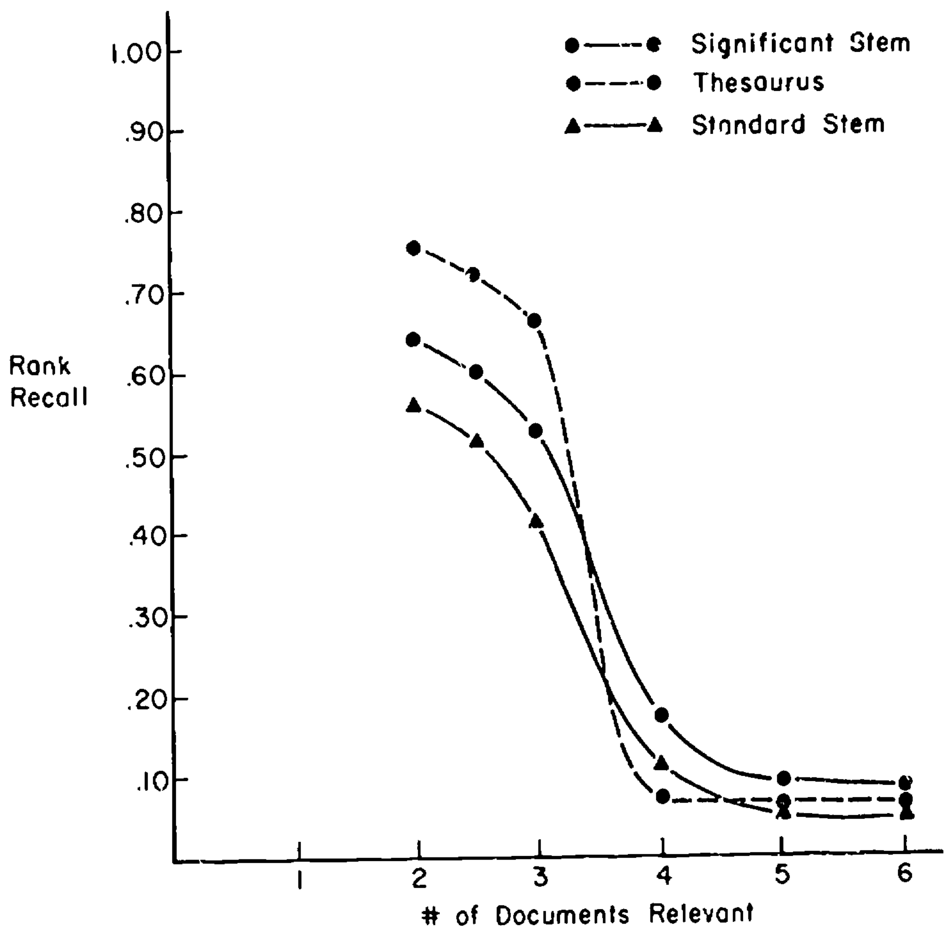
concepts in it, then the added common words in the standard stem query should result in extremely precise retrieval. On the other hand, a very short query in terms of significant concepts would, one supposes, almost have to contain common words if any documents are to be retrieved at all. This hypothesis, however, is not born out by the search results. Graph 8 plots rank recall for the significant and standard stem queries at various query lengths over 42 queries.

Graph 8 indicates that there are indeed differences in the improvement of significant stem over standard stem queries, but there is no easy way to characterize the differences. There are other factors affecting retrieval, such as the number of documents relevant to the query. For example, with a very short query and few relevant documents, common words would be more necessary than if there are a lot of relevant documents. Thus the only fact shown by Graph 8 is that retrieval can vary with the length of the query; the best recall occurs at the average number of significant concepts, which is roughly six.

G) Effect of Query Generality on Search Performance

Remaining is the question of whether it is wise to forget about using a thesaurus with synonyms, since removing common words alone improves stem retrieval. Certainly the recall-precision graphs indicate that precision suffers with the thesaurus, particularly at low recall and document levels. In many cases, then, it appears that synonyms retrieve more non-relevant documents than a dictionary without synonyms.

Graph 9, however, indicates that the picture for the thesaurus is not all that black. This graph shows, for all three dictionaries, rank recall plotted against the number of documents relevant to the query, holding query length constant; when query generality is low, the thesaurus performs best.



Rank Recall vs. # Documents Relevant
(Queries with 6 Significant Concepts)

Graph 9

Using a thesaurus improves the chances of those one or two relevant documents being retrieved, whereas the signficant stem query may fail to correlate with any of the relevant documents. When there are many relevant documents, however, a thesaurus loses its edge because at least one of the relevant documents is likely to be retrieved by any of the queries, and the thesaurus synonyms serve only to retrieve a large amount of non-relevant items.

H) Conclusions of the Global Analysis

The general conclusions which may be drawn from this global analysis are as follows:

1) If one is interested in precision, it is definitely wise to remove common words from the query and document vectors.

2) If one is interested in a high recall ceiling during a full search, one should use a thesaurus. The thesaurus has better ultimate recall than does stem alone, indicating that synonyms retrieve better than common words do.

3) If there are few documents relevant to a query, one should use a thesaurus. Keen reaches much the same conclusion, saying that "for users needing high precision with only one or two relevant documents, the thesaurus is little better than stem on IRE-3, but in CRAN-1 and ADI, a larger superiority for the thesaurus is evident." [2] (CRAN-1 is the same collection as is being used here.) It is possible that while synonyms are useful in the Cran-200 and ADI collections, in other collections synonyms would not be required even for high recall.

4) If there are many relevant documents to a query, it is just as good and perhaps better to remove both common words and synonyms from the query and document vectors.

4. Analysis of Search Performance

Having reached some conclusions on the basis of overall statistics, it is now appropriate to examine the reasons for these results by looking at some specific queries.

The overall averages presented in section 3 indicate the general superiority of the significant stem dictionary over the standard stem dictionary. At all recall (and document) levels, the significant stem has greater precision than does the standard stem. The reason for this improvement in performance can be seen by inspection of Query 36 (Figure 3).

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
37	1	.4234	1	.5292	1	.4889
35	2	.2413	2	.3111	2	.3651
36	7	.1365	4	.2046	6	.2614
34	14	.1064	5	.1519	5	.2505
Rank Sum	.4167		.8333		.7143	
Log Precision	.4503		.8615		.7762	
Norm Recall	.8941		.9974		.9949	
Norm Precision	.7843		.9716		.9491	

Query 36

Figure 3

The standard stem query has two more terms in it than does the significant stem query, "determine" and "establish." It can be seen from Figure 3 that removal of these two common words from the query doubles search effectiveness.

All three queries retrieve documents 35 and 37 first; the standard stem query, however, retrieves four non-relevant documents before the third relevant one. Two of these non-relevant documents are retrieved by the query word "determine" while the other two are retrieved simply because they are short and contain one query term each.

Analysis of this query demonstrates two reasons why removing common words is beneficial to retrieval. re is that common words increase the chances of the query's correlating with a non-relevant document simply because that document and the query have the same common words in them. Secondly, inclusion of common words greatly increases the length of the document vectors, but short texts are lengthened relatively less than are long texts. Thus short texts have a decidedly greater chance of a high correlation with the query; having one term in common with the query gives it a disproportionately high correlation when relevancy should not be a function of text length.

Also indicated by the recall-precision curves is the similarity of the significant stem and thesaurus retrieval, with the significant being slightly better in general. This finding is also borne out by Query 36 (Figure 3), where only two non-relevant documents are retrieved by the thesaurus query, as opposed to the one retrieved by the significant stem query, before a recall level of 1.00 is reached. Interestingly, the short document containing the terms "axial compressor" which was retrieved early by both the stem queries is not one of these two non-relevant documents retrieved early by the thesaurus query; rather, synonyms account for th retrieval of the two non-relevant items. Specifically, the query term "compressor" appears only once in the two non-relevant documents, while the synonym "impeller" appears seventeen times, giving them a high correlation with the thesaurus query.

Query 36 thus demonstrates why synonyms can degrade precision; "compressor" is a frequently occurring word in the Cran-200 collection and in combination with its synonyms can cause retrieval of a number of non-relevant documents. Using stems alone, on the other hand, gives less emphasis to words like "compressor" and more to the group of significant query terms as a whole.

Nevertheless, it is difficult to make hard and fast distinctions between the search precision of thesaurus queries versus significant stem queries. In Query 27 (Figure 4), for example, it is precisely the synonyms which account for the high performance of the thesaurus query. All three versions of Query 27 are identical, except that the thesaurus query, of course, includes synonyms. These synonyms serve to retrieve with relatively high precision the first three relevant documents. Specifically, document 160 does not contain the term "boundary-layer" but it does contain its synonyms "boundary" and "layer" three times each. In this case, the low precision effect of synonyms is offset by the large set of query terms; taken as a whole, the complete set of query terms and their synonyms helps pinpoint the relevant documents more accurately.

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
160	45	.1826	34	.2287	5	.4327
28	43	.1902	46	.2020	8	.3813
56	31	.2105	32	.2297	11	.3750
29	75	.1035	77	.1226	54	.2307
71	62	.1284	57	.1667	71	.1405
161	138	.0309	-	-	166	.0367
Norm Recall	.6796		.3333		.7285	
Norm Precision	.2920		.3754		.4772	
Rank Recall	.0533		.0150		.0623	
Log Precision	.2639		.1692		.3336	

Query 27

Figure 4

The superior correlation of relevant items 28 and 56 with the thesaurus query as opposed to the stem queries is explained by the shorter thesaurus document vector lengths (Figure 5).

Document	Thesaurus Length	Significant Stem Length
28	57	63
56	26	27

Length of Relevant Document Vectors for Query 27

Figure 5

Similarly, the significant stem is more precise than the standard stem because significant stem document vectors are shorter, giving higher weights to their significant terms.

Search results in this study corroborate the findings of past workers that the thesaurus is better than the standard stem dictionaries. The results also indicate that much of this difference may well be attributable to the lengthy common word list of the thesaurus. In Query 36 (Figure 3), for example, the improvement of the thesaurus query over the standard stem query is due more to the removal of common words than to synonyms.

The same improvement can be seen in Query 7 (Figure 6) where the thesaurus results in much better retrieval than the standard stem query. All three queries retrieve the same two relevant and the same non-relevant documents in the first three recovered. After that, however, the next relevant document is found in ranks 11, 13, and 41 for the significant stem, thesaurus, and standard stem queries, respectively. This difference in retrieval is clearly due to the removal of common words, since the two dictionaries with the long common word list ranked about the same. Synonyms

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
41	2	.4042	1	.4914	1	.4762
90	3	.3175	3	.3536	3	.3859
42	41	.1459	11	.2176	13	.2572
72	53	.1279	47	.1211	35	.1918
95	60	.1200	70	.0773	44	.1672
Norm Recall	.8523		.8800		.9169	
Norm Precision	.5944		.6856		.7130	
Rank Recall	.0943		.1136		.1563	
Log Precision	.3528		.4129		.4351	

Query 7

Figure 6

contribute very little to the high precision in the initial retrieval stages.

Results indicate, however, that at the higher recall levels, the thesaurus is superior. This is shown in Query 7 (Figure 6) where the last two relevant documents are retrieved much faster by the thesaurus query than by either of the two stem queries. The reason for this is primarily the shorter document lengths of the thesaurus vectors, and secondarily the synonym "coefficient" is matched with the query term "derivative" in one case. (Shorter document length also explains the faster retrieval of 72 by the significant stem than by the standard stem.) In the case of document 95, however, the standard dictionary works better than the significant stem dictionary because the common terms "comparison" and "number" combined with the significant "mach" boost the document-query correlation of 95.)

That the significant stem dictionary has severe short-comings in the lower correlation, high recall, ranges is without doubt. This degradation in recall is not fully reflected by the recall-precision graphs, though it is seen in the normalized global statistics (Figure 1).

The main explanation for this phenomenon appears to be that the significant stem vectors, with neither common words nor synonyms in them, have a good chance of "missing" a relevant document altogether. Query 23 (Figure 7) demonstrates this in that one of the two relevant documents does not correlate at all with the significant stem query.

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
143	3	.2197	5	.1257	10	.1991
148	13	.1346	-	-	5	.2683
Norm Recall	.9672		.4899		.9697	
Norm Precision	.6999		.3722		.6748	
Rank Recall	.1875		.0146		.2000	
Log Precision	.1892		.1003		.1772	

Query 23

Figure 7

In this query, Item 148 has none of the significant query terms. It does, however, contain the synonyms "impeller" and "Compressor" for the query term "pump," and it also contains "method," a common term found in the standard stem query. (It should be noted that Document 148 is picked up after feedback for the significant stem query.)

While both common words and synonyms are useful for retrieval at high recall levels, synonyms are superior in this respect. In Query 3 (Figure 8) the thesaurus is the only dictionary of the three which achieves 100% recall during the full search.

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
57	3	.2134	3	.2889	8	.3303
31	24	.1331	14	.1862	13	.2476
30	16	.1486	21	.1795	20	.2182
32	9	.1825	10	.2102	23	.2001
4	18	.1450	19	.1827	25	.1876
33	-	-	-	-	124	.0441
Norm Recall	.7861		.7887		.8351	
Norm Precision	.5681		.5724		.5132	
Rank Recall	.0778		.0787		.0986	
Log Precision	.3774		.3797		.3497	

Query 3

Figure 8

The only reason that document 33 is retrieved by the thesaurus is that it contains the term "high-pressure-ratio" which matches "pressure" in the thesaurus query. Even the five extra terms added to the standard stem dictionary query fail to retrieve this last relevant item.

It is interesting to note here that while recall is superior for the thesaurus in Query 3, precision is not. The synonyms, as noted above, retrieve many non-relevant documents, and here more so than even common words do. Once again, the rule that high recall means low precision seems to be borne out.

Although the significant stem fails to achieve a 100% recall ceiling more often than both the other dictionaries, there are cases when high precision, low recall, and feedback can be effectively used to achieve high precision and high recall. One case of this is Query 1 (Figure 9) where so many non-relevant items are retrieved by the thesaurus and the standard stem that feedback is impossible because the user sees no relevant documents. Once again, as is typically the case, the thesaurus has the highest recall ceiling but not very precise retrieval.

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
22	29	.0899	1	.2209	33	.1109
21	-	-	-	-	32	.1115
1	-	-	-	-	-	-
	Query 1 after feedback					
22	29	.0899	1	.9796	33	.1109
21	-	-	9	.0955	32	.1115
1	-	-	2	.1996	-	-

Query 1

Figure 9

The significant stem query retrieves only one of the three relevant items (22), but this item is used for positive feedback and in turn retrieves another relevant document (21). No feedback, on the other hand, can be done with the standard stem query (only 22 correlates, and it is in rank 29) or with the thesaurus query (two relevant documents correlate with the query, but are in ranks 32 and 33). Thus query 1 demonstrates that it is not always necessary to have complete recall, at least during the initial search; high precision is more useful if feedback is going to be used.

The feedback recall-precision graphs in Appendix II indicate that this is precisely what happens, since feedback improves the precision of the significant stem much more than the other two dictionaries at the high recall end of the curve.

The <u>effect of query length on precision</u>, where length is the number of significant concepts in the query vectors, does not appear to vary retrieval results in a consistent manner. If a query is worded very specifically, which dictionary used is immaterial (see Query 12, Figure 10). On the other hand, a lengthy query may zero in faster on relevant documents but in the long run retrieves more non-relevant ones.

Relevant Document #	Standard Stem Rank & Corr.		Significant Stem Rank & Corr.		Thesaurus Rank & Corr.	
46	1	.5175	3	.5284	5	.5217
49	2	.4759	2	.5423	2	.7272
48	4	.4308	7	.4558	7	.4937
50	5	.3996	4	.5185	3	.6963
47	6	.3857	5	.4642	6	.5067
51	7	.3776	8	.4082	8	.4660
Norm Recall	.9966		.9931		.9914	
Norm Precision	.9663		.9111		.8950	
Rank Recall	.8400		.7241		.6774	
Log Precision	.8859		.7466		.7137	

Query 12
Figure 10

The length of the query is less important than the number of documents relevant to a query. If there are a lot of documents relevant to a query, it is often better to use a narrow query first (no common words or synonyms) and then use feedback to retrieve the remaining relevant items. In Query 16 (Figure 11) the thesaurus has the highest recall ceiling in the full search, but at the same time retrieves so many non-relevant that only one relevant item is available for feedback. The standard stem does not have quite a high recall ceiling and also has only one document in the top five for feedback. The significant stem, however, retrieves two relevant in the top five and so feedback is more effective (the total of relevant document ranks after feedback is the least for the significant stem query).

Relevant Document Number	Standard Stem Full	Standard Stem Feed-back	Significant Stem Full	Significant Stem Feed-back	Thesaurus Full	Thesaurus Feed-back
102	2	1	2	1	2	1
84	9	37	5	2	20	25
83	7	5	9	3	11	5
81	-	2	-	4	70	2
80	15	3	15	5	27	3
82	-	16	-	6	-	13
193	18	18	21	14	9	4
67	24	31	22	38	46	41
85	-	50	-	41	-	33
Sum of ranks after feedback		163		114		127

Query 16, Full Search and Feedback Rankings

Figure 11

It seems obvious, then, that <u>an extensive common word list is helpful in retrieval</u>, particularly if precision is desired. If one wishes to improve upon a standard stem dictionary, the first thing he should do is to find a good, extensive common word list. After that, additional improvement may be gained (in recall, particularly) by grouping some of the dictionary terms into concept clssses. Doing it the other way around can be disastrous, however, as is seen in Query 19 (Figure 12).

Relevant Document #	Standard Stem Rank	Standard Stem Corr.	Significant Stem Rank	Significant Stem Corr.	Thesaurus Rank	Thesaurus Corr.
123	19	.2016	3	.3636	15	.2303
125	20	.1990	5	.3079	21	.2052
122	6	.2490	6	.2814	18	.2107
124	47	.1254	18	.1866	62	.1375
Norm Recall	.8354		.9719		.8648	
Norm Precision	.5327		.7658		.4667	
Rank Recall	.1087		.3125		.0862	
Log Precision	.2744		.4300		.2489	

Query 19

The significant stem dictionary here is clearly the best and the thesaurus is the worst. In Query 19, there are eight significant terms which in themselves result in good retrieval (as indicated by the performance of the significant stem query). In addition to these eight terms, there are five common terms in the standard stem query, causing it to retrieve five non-relevant items before the first relevant one. Figure 13 shows how the significant terms can be overwhelmed by insignificant terms.

Document	94	86	64	25	148	122 R
signif. terms, in all queries	planform rectangular wing	analytic flow oscillate rectangular wing	planform wing	analytic flow transonic	flow	flow oscillate transonic wing
common terms, in stand. stem only	determine general method possible	determine general method	general method	determine general method	determine general method	method

Terms (and Number of Occurrences) Appearing in Top 6
Documents Retrieved by Standard Stem Query 19

Figure 13

The thesaurus query vector for some reason contains three of the common terms added to Query 19; it does even worse than the stem dictionaries because synonyms compound the difficulties of common words. The thesaurus query thus retrieves 14 non-relevant documents before finding the first relevant one. The query terms "oscillater' and "planform" both belong to relatively large synonym classes.

5. Conclusions

The main conclusion of this study in the area of dictionary construction is that careful construction of common word lists is at least as important as grouping concepts into synonym classes. This is an important result since it should be earier to construct common word lists automatically than to construct synonym classes automatically.*

This study, in addition, has relevance to areas other than dictionary construction. For example, a fair amount of work is being done in the area of automatic document vector modification, which in part involves dropping "unimportant" concepts from the vectors (i.e., concepts infrequently used in queries). Since the common word list used in this study also contains infrequent words whereas the standard stem dictionary merely includes them as regular words, there is an opportunity in local analysis of these search runs to determine the effect of infrequently used words on retrieval. In particular both Query 6 and Query 1 in some of their versions included an infrequent word not in the other versions. In neither case, did this infrequent word affect retrieval except lower correlations by lengthening the query vector.

Another area in which this study is relevant is in scatter storage schemes for dictionary lookups [3]. This scheme can offer improvements in efficiency but thesaurus-type dictionaries are difficult to handle. One has to make a two-step mapping in order to get to the synonym class from the original query or document term; common words, on the other hand, can

---

* Work is being done in automatic synonym construction or has been done [1]. For these algorithms to work, however, common words probably have to be removed first, anyway.

be handled easily enough. Therefore having determined that a standard stem dictionary can be considerably improved by removing some words into the common word list, it would be better to implement this improvement in the storage scatter scheme than it would be to implement the improvement involving concept classes.

Finally, this project carries out a suggestion made by Keen [2] that is the "five rules" of thesaurus construction are to be really evaluated, several different versions of a single dictionary would have to be made and tested. In the course of this study, a new dictionary is created, one which uses the frequency rules but not the grouping rules. Thus the importance of rules dealing with word frequency versus rules about synonym classes is established. It is just as important to be careful in constructing the common word list as in constructing the thesaurus. However, it is probably easier to follow the rules for common work list construction since common words are more systematic than synonyms are.

6. Further Studies

This investigation raises a few issues which were not settled, and which may prove interesting for further study:

1) The work presented in this paper is of course not conclusive for collections other than the Cran-200. The first extension of this experiment, then, would be to perform a similar common word analysis on other collections. One reason for the apparent good performance of the significant stem dictionary is that the Cran-200 thesaurus is not that much better than the standard stem dictionary in the first place.

2) The current Cran-200 collection still contains a fair number of common words in the thesaurus vectors although these same words have been marked common in the thesaurus itself. This could also explain the lack of performance of the thesaurus as compared with the significant stem dictionary. Thus a new look-up run should be made on the Cran-200 collection using the current version of the thesaurus to generate vectors without so many common words in them.

3) It would be interesting to determine more precisely the influence of infrequent words on retrieval.

4) More careful analysis of feedback results from this investigation should be made.

References

[1] R. T. Dattola and D. M. Murray, "An Experiment in Automatic Thesaurus Construction," Report No. ISR-13 to the National Science Foundation, Section VIII, Cornell University, Department of Computer Science, 1968.

[2] E. M. Keen, "Thesaurus, Phrase & Hierarchy Dictionaries," Report No. ISR-13 to the National Science Foundation, Section VII, Cornell University, Department of Computer Science, 1968.

[3] D. M. Murray, "A Scatter Storage Scheme for Dictionary Lookups," Report No. ISR-16 to the National Science Foundation, Section II, Cornell University, Department of Computer Science, 1969.

Appendix I

Some query vectors using the standard stem, significant stem and thesaurus

Query	Standard Stem		Significant Stem		Thesaurus	
	4116	gas	863	gas	226	gas
	5087	kinetic	1139	kinetic	118	kinetic
	2086	Chapman-Enskog				
1						
	2576	detail			275	results, solution
	7296	rigorous				
	9083	theo-			33	theory
	1553	bound-	253	boundary	394	boundary
2	2463	cylinder	484	cylinder	158	cylinder
	3392	flow	777	flow	389	flow
	5171	layer	1178	layer	394	layer
			1441	non-circular	151	non-circular
	2666	dissociate	568	dissociate	89	dissociate
	3137	enthalpy	656	enthalpy	294	enthalpy
	3479	free	822	free	11	free
	4407	hypersonic	977	hypersonic	57	hypersonic
	6625	press-	1690	pressure	386	pressure
	8248	simulate	2019	simulate	194	simulate
3	8546	stream	2202	stream	414	stream
	9306	tunnel	2419	tunnel	190	tunnel
	9725	wind	2588	wind	190	wind
	4305	high			47	high
	6558	possible				
	7113	realize			521	real, practical
	7249	respect				
	8234	significant				
	2447	current	477	current	332	current
	2609	differ-	547	difference	105	difference
	3035	effect	610	effect	388	effect
	4258	heat	906	heat	276	heat
	5168	law	1176	law	270	law
	8465	stagnation-point	2152	stagnation-point	134	stagnation-point
4	9238	transfer	2389	transfer	251	transfer
			2534	viscosity-temperature		
	9618	vortice	2548	vortic-	281	vortic-
	1218	analyses			31	analyses
	1334	assume			17	assume
	2641	discrepancy				
	6652	prime			44	prime ?
	7257	result				

Query	Standard Stem		Significant Stem		Thesaurus	
	4407	hypersonic	977	hypersonic	57	hypersonic
	5171	layer	1178	layer	394	layer
	5239	line-	1217	linear	288	linear
	7289	reynold-	1866	reynolds	362	reynolds
	8184	shock	1982	shock	387	shock
5			2534	viscosity-temperature		
	1218	analyses			31	analyses
	1334	assume			17	assume
	5321	low			46	low
	6196	number			384	number
	8358	solution				
	1388	axial	164	axial	185	axial
	2226	compress-	372	compressor	202	compressor
	5090	kink	1140	kink	242	kink
6	5239	line	1216	line	68	line
	5594	multi-stage	1402	multi-stage		
	8665	surge	2256	surge	149	surge
	3248	explain				
	1102	aerodynamic	39	aerodynamic	137	aerodynamic
	2551	derivatives	525	derivative	429	derivative
	4407	hypersonic	977	hypersonic	57	hypersonic
	5348	mach	1269	mach	392	mach
	5441	measure	1319	measure	32	measure
7						
	2207	compare				
	6196	number			384	number
	9086	theoretic			36	theoretical
	9764	work				
	1102	aerodynamic	39	aerodynamic	137	aerodynamic
	2551	derivatives	525	derivative		
	3285	facility	715	facility	207	facility
	5441	measure	1319	measure	32	measure
8	7353	run-	1899	running	289	running
	8208	short	2003	short	53	short
	9169	time	2356	time	9	time
	1084	adopted				
	1377	avail				
	5479	method				
	1107	aerofoil	44	aerofoil	197	aerofoil
	2370	correct-	439	correction		
	5582	mount	1385	mount	55	mount
9	9306	tunnel	2419	tunnel	190	tunnel
	9330	two-dimensional	2[illegible]36	two-dimension-	104	two-dimension-
	9727	wind-tunnel	2589	wind-tunnel	190	wind-tunnel

Query	Standard Stem		Significant Stem		Thesaurus	
	3392	flow	777	flow	389	flow
	7019	quasi-conical	1761	quasi-conical	157	quasi-conical
	8480	state	2163	state	26	state
10						
	6621	present				
	9083	theo-			33	theory
	3392	flow	777	flow	389	flow
	5128	laminar	1152	laminar	94	laminar
	5543	model	1367	model	194	model
	6019	nature-	1410	natural	297	natural
	6086	parameter	1580	parameter	271	parameter
11	9242	transit-	2392	transition	394	transition
	9306	tunnel	2419	tunnel	190	tunnel
	9316	turbulent	2426	turbul-	286	turbul-
	9725	wind	2588	wind	190	wind
	4566	influence			249	influence
	1060	act-	24	action	250/249	action
	1139	air	63	air	165/228	air
	1192	altitude	92	altitude	184	altitude
	1348	atmosphere	151	atmosphere	228	atmosphere
	2712	drag	588	drag	135	drag
	4273	height	918	height	184	height
12	6284	orbit	1534	orbit	460	orbit
	8024	satellite	1913	satellite	318	satellite
	8031	scale	1915	scale	43	scale
	2334	contract				
	9536	vary			239	adjust
	2543	delta	516	delta	159	delta
	3392	flow	777	flow	389	flow
	8438	speed	2118	speed	253	speed
	8682	sweptback	2268	sweptback	50	sweptback
13	9035	tapered	2298	taper-	498	taper-
	9253	transonic	2398	transonic	296	transonic
	9755	wing	2592	wing	223	wing
	2609	differ			239	adjust

Query Vectors for Three Dictionary Types

Run 0 — 42 Queries (Plus 0 Nulls) — Wordstem Feedback = Standard
A Full Search with One Iteration of Feedback Using Word Stem Dictionary

Run 1 — 42 Queries (Plus 0 Nulls) — Cranmine Feedl = Sig Stem
Full Search with One Iteration of Feedback using Stems with Common Words

Run 2 — 42 Queries (Plus 0 Nulls) — Thesaurus Feedback
A Full Search with One Iteration of Feedback

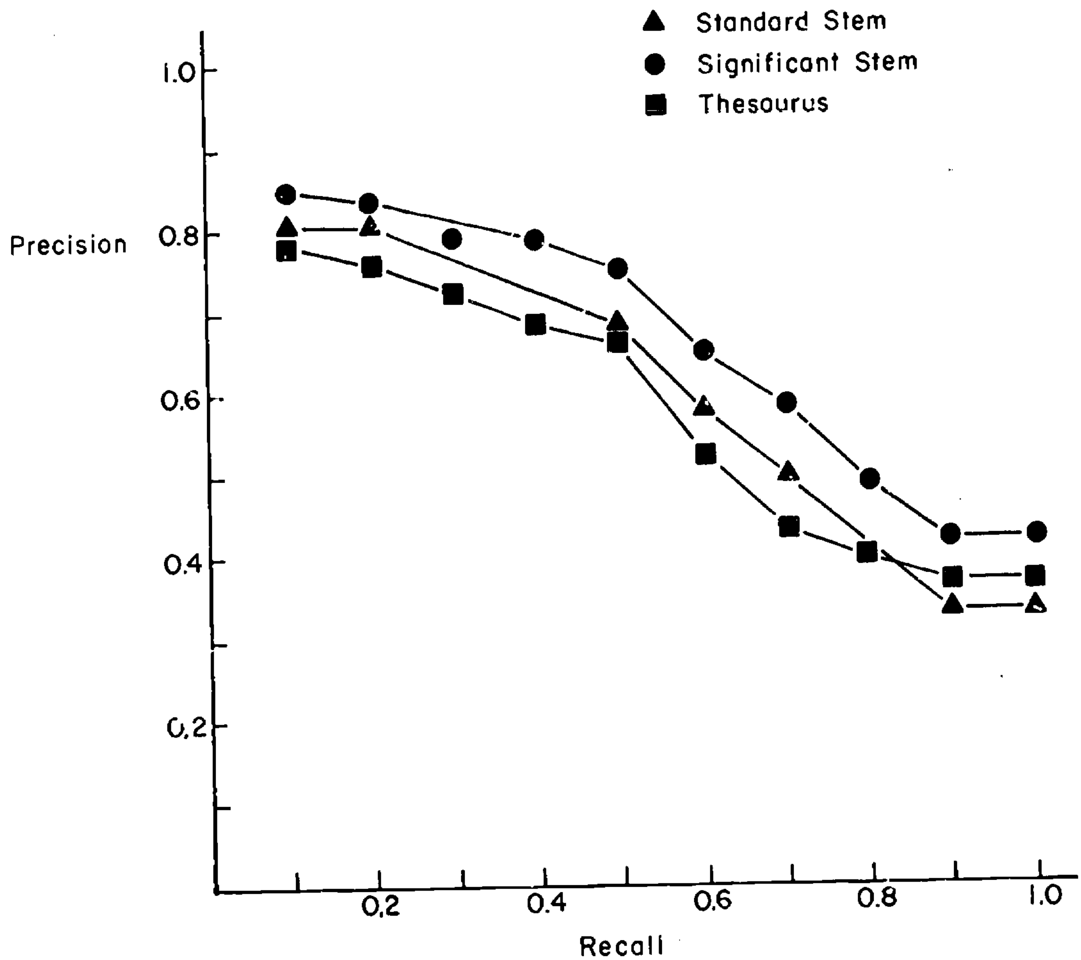
	Run 0		Run 1		Run 2	
Recall	NQ	Precision	NQ	Precision	NQ	Precision
0.0	0	0.8098	0	0.8484	0	0.7783
0.05	0	0.8098	0	0.8484	0	0.7783
0.10	1	0.8098	1	0.8484	1	0.7783
0.15	8	0.8098	8	0.8484	8	0.7664
0.20	21	0.8098	21	0.8246	21	0.7521
0.25	31	0.8098	30	0.8067	31	0.7291
0.30	31	0.7881	30	0.7885	31	0.7162
0.35	32	0.7710	32	0.7862	33	0.6983
0.40	32	0.7110	32	0.7862	33	0.6881
0.45	32	0.6810	32	0.7455	33	0.6597
0.50	40	0.6810	40	0.7455	41	0.6597
0.55	40	0.5883	40	0.6612	41	0.5503
0.60	40	0.5759	40	0.6479	41	0.5117
0.65	40	0.5234	40	0.6241	41	0.4757
0.70	40	0.4916	40	0.5799	40	0.4351
0.75	40	0.4698	40	0.5509	40	0.4347
0.80	40	0.4043	37	0.4831	39	0.3953
0.85	40	0.3736	34	0.4419	38	0.3734
0.90	40	0.3486	34	0.4278	38	0.3593
0.95	40	0.3486	34	0.4278	38	0.3580
1.00	41	0.3486	35	0.4278	39	0.3580
Norm Recall		0.6955		0.9011		0.9045
Norm Precision		0.7647		0.7999		0.7597
Rank Recall		0.4082		0.4997		0.4207
Log Precision		0.6001		0.6647		0.5885

Symbol Keys: NQ = Number of Queries used in the average not dependent on any extrapolation.
Norm = Normalized.

Recall Level Averages

Appendix 2

Recall Revision Results



Recall Level Graph

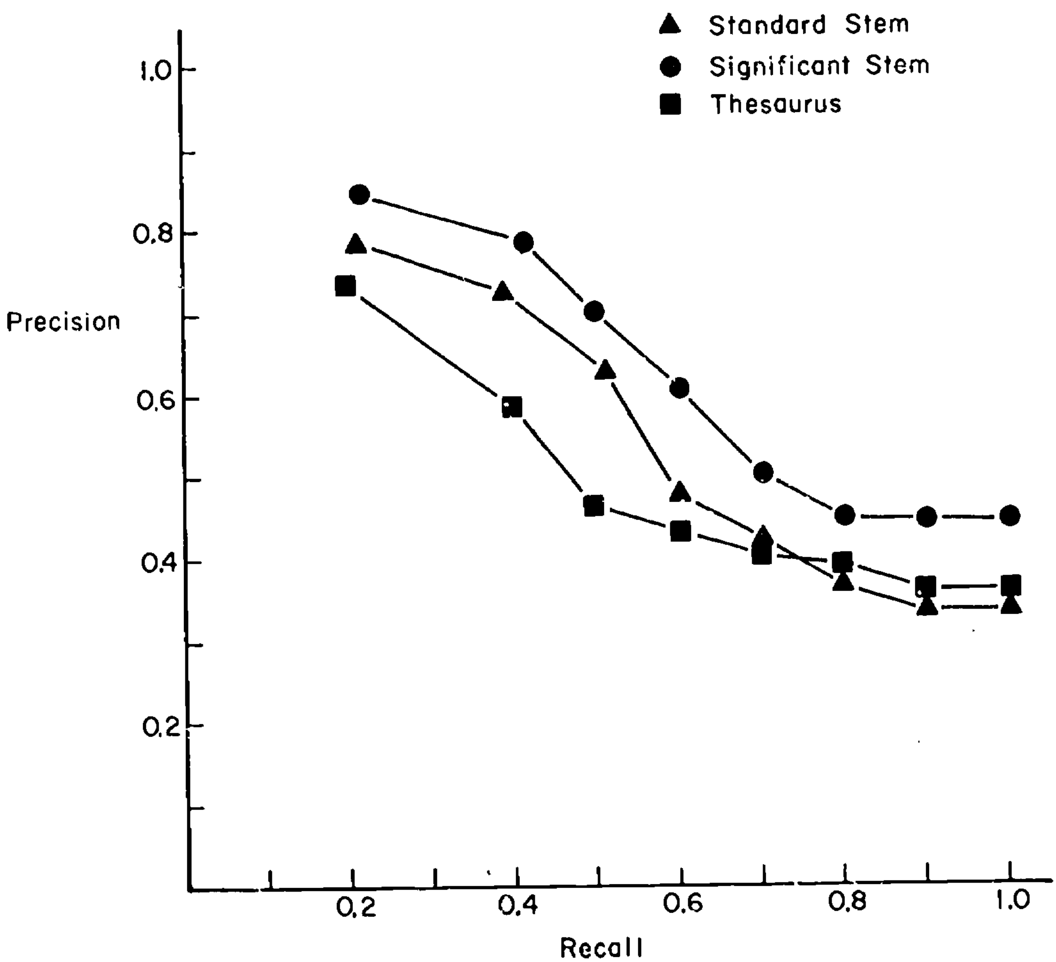
Run 0 — 42 Queries (Plus 0 Nulls) — Wordstem Feedback = Standard
A Full Search with One Iteration of
Feedback Using Word Stem Dictionary

RUN 0

Rank	NR	CNR	NQ	Recall	Precision
1	33	33	42	0.2266	0.7857
2	27	60	41	0.3817	0.7262
3	17	77	36	0.4565	0.6667
4	13	90	35	0.5129	0.6190
5	5	95	34	0.5293	0.5571
6	8	103	34	0.5651	0.5278
7	4	107	33	0.5798	0.4955
8	5	112	31	0.5993	0.4789
9	1	113	29	0.6033	0.4584
10	1	114	28	0.6072	0.4436
11	4	118	28	0.6287	0.4379
12	3	121	28	0.6416	0.4313
13	2	123	28	0.6485	0.4238
14	3	126	28	0.6622	0.4191
15	3	129	28	0.6749	0.4150
16	2	131	28	0.6805	0.4093
17	3	134	28	0.6921	0.4069
18	1	135	28	0.6947	0.4015
19	2	137	28	0.7054	0.3980
20	2	139	28	0.7148	0.3948
30	11	150	26	0.7612	0.3702
50	19	169	20	0.8448	0.3531
75	16	185	9	0.9321	0.3514
100	2	187	8	0.9395	0.3491
	11	198			
10.0%	139	139	28	0.7148	0.3948
25.0%	30	169	20	0.8448	0.3531
50.0%	18	187	8	0.9395	0.3491
75.0%	6	193	3	0.9683	0.3484
90.0%	1	194	2	0.9742	0.3483
100.0%	4	198	0	1.0000	0.3486

Symbol Keys: NR = Number of Relevant.
CNR = Cumulative Number of Relevant.
NQ = Number of Queries used in the Average not Dependent on any Extrapolation.
% = Percent of Total Number of items in Collection.

Document Level Averages (1)



Document Level Averages

Run 1 — 42 Queries (Plus 0 Nulls) — Cranmine Feedl = Sig Stem
Full Search with One Iteration of Feed-
back using Stems with Common Words

RUN 1

Rank	NR	CNR	NQ	Recall	Precision
1	35	35	42	0.2405	0.8333
2	28	63	41	0.4146	0.7619
3	18	81	35	0.5011	0.7063
4	12	93	32	0.5479	0.6528
5	9	102	31	0.5848	0.6111
6	8	110	31	0.6170	0.5794
7	5	115	29	0.6393	0.5510
8	5	120	27	0.6594	0.5349
9	3	123	26	0.6772	0.5170
10	2	125	23	0.6868	0.5038
11	2	127	22	0.6941	0.4940
12	4	131	21	0.7128	0.4912
13	4	135	20	0.7273	0.4893
14	2	137	20	0.7329	0.4843
15	2	139	20	0.7448	0.4800
16	2	141	19	0.7525	0.4767
17	1	142	19	0.7555	0.4723
18	1	143	19	0.7603	0.4684
19	0	143	19	0.7603	0.4637
20	1	144	19	0.7642	0.4606
30	12	156	18	0.8064	0.4429
50	20	176	11	0.8885	0.4355
75	6	182	6	0.9216	0.4310
100	4	186	2	0.9397	0.4291
	12	198			
10.0%	144	144	19	0.7642	0.4606
25.0%	32	176	11	0.8885	0.4355
50.0%	10	186	2	0.9397	0.4291
75.0%	2	188	0	0.9504	0.4275
90.0%	0	188	0	0.9504	0.4269
100.0%	10	198	0	1.0000	0.4278

Symbol Keys:
NR = Number of Relevant.
CNR = Cumulative Number of Relevant.
NQ = Number of Queries used in the Average not Dependent on any Extrapolation.
% = Percent of Total Number of Items in Collection.

Document Level Averages (2)

Run 2 — 42 Queries (Plus 0 Nulls) — Thesaurus Feedback
A Full Search with One Iteration of Feedback

RUN 2

Rank	NR	CNR	NQ	Recall	Precision
1	31	31	42	0.2099	0.7381
2	24	55	41	0.3541	0.6667
3	10	65	36	0.3888	0.5714
4	15	80	36	0.4592	0.5536
5	6	86	34	0.4811	0.5060
6	4	90	34	0.5012	0.4663
7	8	98	34	0.5399	0.4515
8	9	107	33	0.5807	0.4452
9	6	113	29	0.6138	0.4389
10	2	115	28	0.6232	0.4254
11	6	121	27	0.6506	0.4239
12	3	124	25	0.6625	0.4186
13	4	128	25	0.6787	0.4160
14	1	129	25	0.6821	0.4087
15	2	131	24	0.6928	0.4047
16	1	132	24	0.6975	0.3998
17	3	135	24	0.7142	0.3982
18	2	137	23	0.7249	0.3958
19	2	139	23	0.7327	0.3936
20	3	142	23	0.7426	0.3929
30	15	157	22	0.7990	0.3777
50	18	175	15	0.8886	0.3662
75	10	185	10	0.9331	0.3616
100	0	185	10	0.9331	0.3583
	13	198			
10.0%	142	142	23	0.7426	0.3929
25.0%	33	175	15	0.8886	0.3662
50.0%	10	185	10	0.9331	0.3583
75.0%	9	194	2	0.9774	0.3580
90.0%	0	194	1	0.9774	0.3576
100.0%	4	198	0	1.0000	0.3580

Symbol Keys:
NR = Number of Relevant.
CNR = Cumulative Number of Relevant.
NQ = Number of Queries used in the Average not Dependent on any Extrapolation.
% = Percent of Total Number of Items in Collection.

Document Level Averages (3)

VI. Negative Dictionaries

K. Bonwit and J. Aste-Tonsmann

Abstract

A rationale for constructing negative dictionaries is discussed. Experimental dictionaries are produced and retrieval results examined.

1. Introduction

Information retrieval often involves language processing, and language processing frequently leads to language analysis. When the information initially appears in natural language form, it is desirable to perform some sort of normalization at the beginning of the analysis. A system often used in practice assigns keywords, or index terms, to identify the given information items. Dictionaries, listing permissible keywords and their definitions, are employed in this process. Sometimes, a negative dictionary is also used, to identify those terms which are not to be assigned as keywords.

Various types of positive dictionaries, their construction and uses, have been discussed elsewhere [1, 2, 3]. The question of the negative dictionary, or, what to leave out, is a fuzzy one. It is generally agreed that "common function words", such as "and", "or", "but", which add to the syntax but not the semantics of a sentence, should be dropped for the purposes of information retrieval. Other words at the extreme ends of the frequency distribution cause a problem. For example, "information" and "retrieval" might appear in nearly every document of a collection on that subject (high frequency); if included as keywords, they would retrieve every-thing. Conversely, if only one document discusses "microfiches" (low frequency), and that word does not constitute one of the permissible keywords, that document may never be retrieved. As with most information retrieval problems, the goals of the system, either high recall or high precision, will determine how many words are to be included. In the SMART system, a standard list of 204 "common English words" is used as a negative dictionary for all collections.

The general procedure used for dictionary construction consists in producing a concordance of the document collection with a frequency count, and including in the negative dictionary rare, low frequency words, common high frequency words, and words which appear in only nonsignificant contexts, such as "observe" in "we observe that . . ." This process requires the choice of frequency cutoff points, and a definition of the notion of "nonsignificance". It presumes _a priori_ that such deletions will not effect retrieval results too considerably. A preferable system would be one that produces a negative dictionary of those terms which can be shown to detract from retrieval efficiency, or at least, not to affect it.

2. Theory

The set of keywords chosen for identifying documents constitutes the _index language_. The number and type of words included will control the _specificity_ of the index language. Keen states [3] that

> "a dictionary which provides optimum specificity for a given test environment will exhibit a precision versus recall curve that is superior to all others probably over the whole performance range."

The purpose of this report is to exhibit a means of measuring specificity, and to show how a negative dictionary can be constructed to optimize index language specificity.

The aim of a negative dictionary is to delete from the index language all words which do not distinguish, and leave only those words which discriminate, among the documents. If the documents are considered as points in a vector space, with the associated identifying keywords as coordinates, then documents containing many of the same keywords will be relatively close together. If all keywords are permitted, then the documents will all cluster in the subspace defined by the common words; on the other hand, if only discriminators are permitted, the document space will "spread out", since each discriminator separates the space into those documents it identifies and those it does not.

The standard method for measuring "closeness", or correlation, of two document vectors $\underline{v}$ and $\underline{w}$ is the cosine:

$$\cos(\underline{v},\underline{w}) = \frac{\sum v_i \cdot w_i}{\sqrt{\sum v_i^2 \cdot \sum w_i^2}}$$

where $v_i\,(w_i)$ is the weight of the i^{th} keyword in document $\underline{v}$ $(\underline{w})$, and the sums run over all possible keywords.

The "compactness" ("closeness together") of the points in the document space can be measured as follows:

1) find the centroid $\underline{c}$ of all the document points, that is,

$$c_i = \frac{1}{N} \cdot \sum_{j=1}^{N} v_{ij}$$

where v_{ij} is the weight of the i^{th} keyword in document j, and N is the total number of documents;

2) find the correlation of each document with the centroid, i.e., cos $(\underline{c},\underline{v}_j)$, for all documents j;

3) define the <u>document space similarity</u>, Q, as:

$$Q = \sum_{j=1}^{N} \cos (\underline{c},\underline{v}_j)$$

Q has values between O and N, higher values indicating more similarity among documents. The value O is never obtained since $\underline{c}$ is a function of the other vectors, and the value N is obtained only if all the documents are identical to the centroid. Normalized Q, i.e. Q/N, is just the average document-centroid correlation (though this value is never calculated in the work which follows).

By calculating Q, using the terms provided by differing index languages, it is possible to measure and compare the specificity of these languages — a language is more specific the lower its Q. The question remains how to discover the optimal Q that will give the superior recall-precision curve described by Keen.

To see what happens when a single keyword is deleted, let Q_i be defined as Q calculated with the i^{th} term deleted (i.e., v_{ij} left out of all calculations, for all documents j). Then, $|Q - Q_i|$ measures the change in document space similarity due to the deletion of term i. If $Q_i > Q$, the document space is more "bunched up", more similar, when term i is deleted, or term i is a discriminator. Conversely, if $Q_i < Q$, deletion of term i causes the space to "spread out", to be more dissimilar, and deletion of term i may aid in retrieval. In the same way, Q_I is defined for a set of terms,

$I = \{i_1, i_2, \ldots, i_n\}$. That is, Q_I measures the document space similarity when all the terms in set I have been deleted from the index language.

Since deletion of discriminators raises Q and deletion of non-discriminators lowers Q, some optimal set of terms I_{min} should exist such that $Q_{I_{min}}$ is minimal. It still remains to be shown that the index language consisting of the set of keywords remaining when the set I_{min} is deleted from the total collection of keywords will be optimal in the sense of Keen. If the total set of keywords is $K = \{i_1, i_2, \ldots, i_t\}$, and $I_{min} = \{i_1, \ldots, i_{min}\}$, $min \leq t$, then Figure 1 describes what should happen to Q as terms are successively deleted from K (a point (i_j, Q) represents $Q_{\{i_1, \ldots, i_j\}}$, i.e., Q for the index language given by $K - \{i_1, \ldots, i_j\}$). As non-discriminators are deleted, the document space spreads out and Q goes down to its minimum. Then as discriminators are deleted, documents that were distinguished are coalesced, the document space draws together, and Q goes up (until all documents are identically null).

It may be hypothesized that retrieval will follow the same pattern. That is, using some method of retrieval evaluation, the best results will occur at $Q_{i_{min}}$, and as Q increases, retrieval "goodness" will decrease. One measure of retrieval effectiveness is the rank of the last relevant document retrieved. If N_r is the average rank (over a group of queries) of the last relevant document retrieved, then assuming retrieval follows Q, N_r versus i will be as in Figure 2. As non-discriminators are deleted (i_1 to i_{min}), it is easier to find the relevant documents, and N_r goes down until i_{min} is reached. At that point discriminators begin to be lost, the document space closes up, relevant documents move closer to non-relevant,

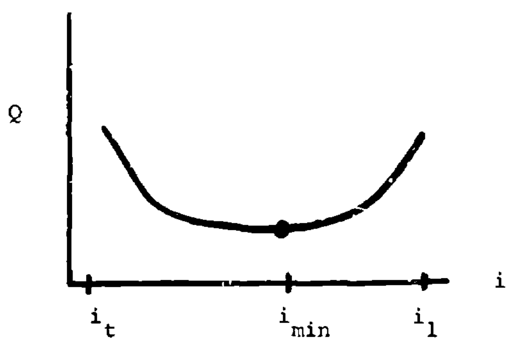


Figure 1

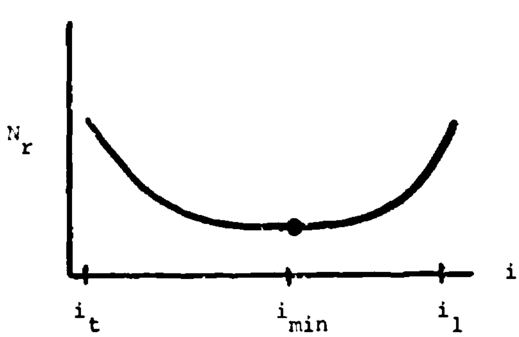


Figure 2

more non-relevant are retrieved along with relevant, and N_r goes back up.

3. Experimental Results

The ADI abstracts collection is used as a base for t sting the above predictions about the Q and N_r curves. The full (no common words deleted) vectors and the accompanying word stem dictionary are used. The dictionary terms are ranked twice:

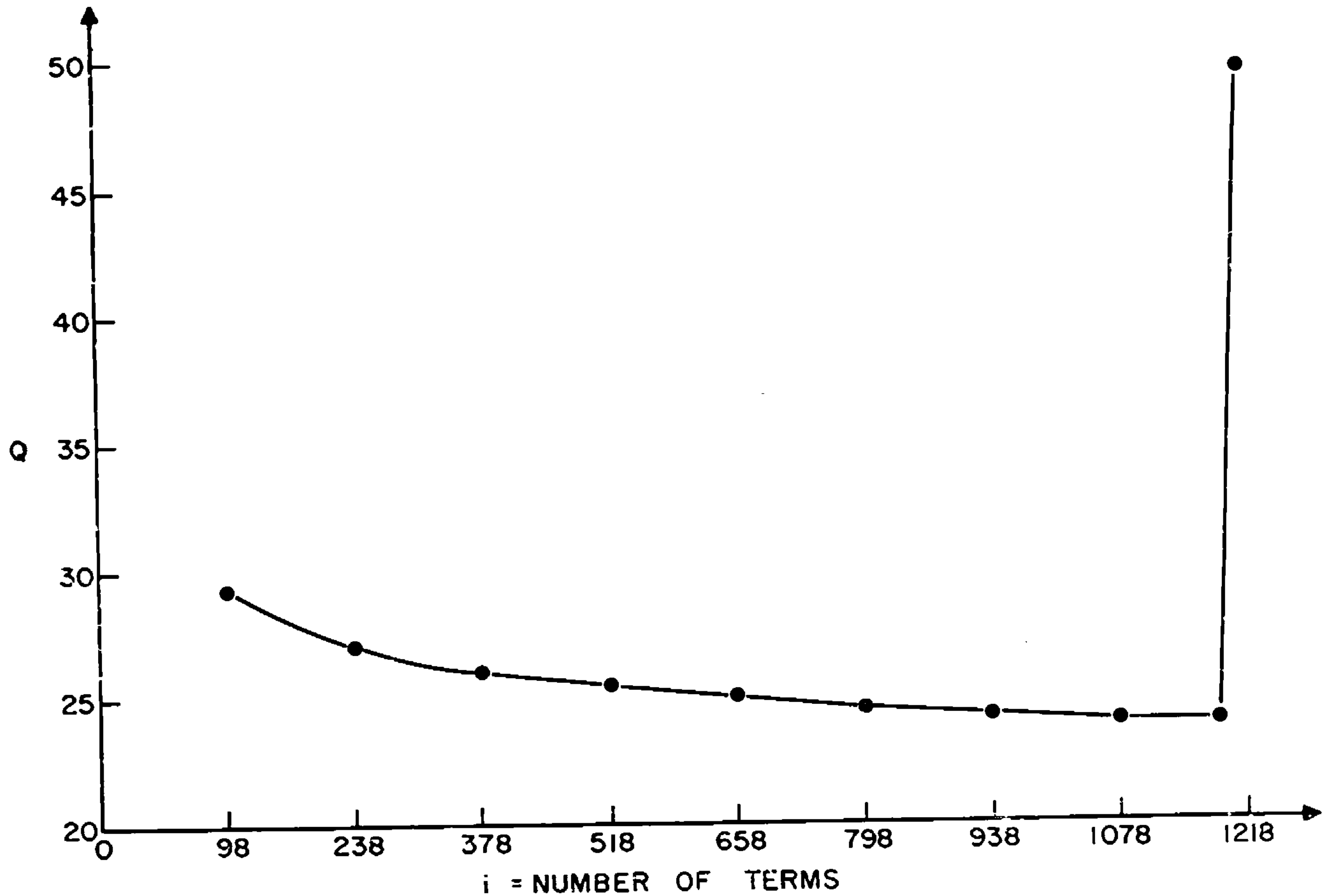
a) in order of increasing Q_i, i.e., with the supposed discriminators at the end of the list;

b) in order of decreasing frequency of occurrence (number of documents appeared in), with the least frequent terms at the end.

Since the ADI collection contains 1218 keywords, only every 28th (an arbitrary number) point of the curves is considered, i.e., what happens when terms 1-28, 1-56, 1-84, . . . are deleted (using the orderings above). At the selected cutoffs, query searches are performed, and the corresponding Q_I's and N_r's calculated.

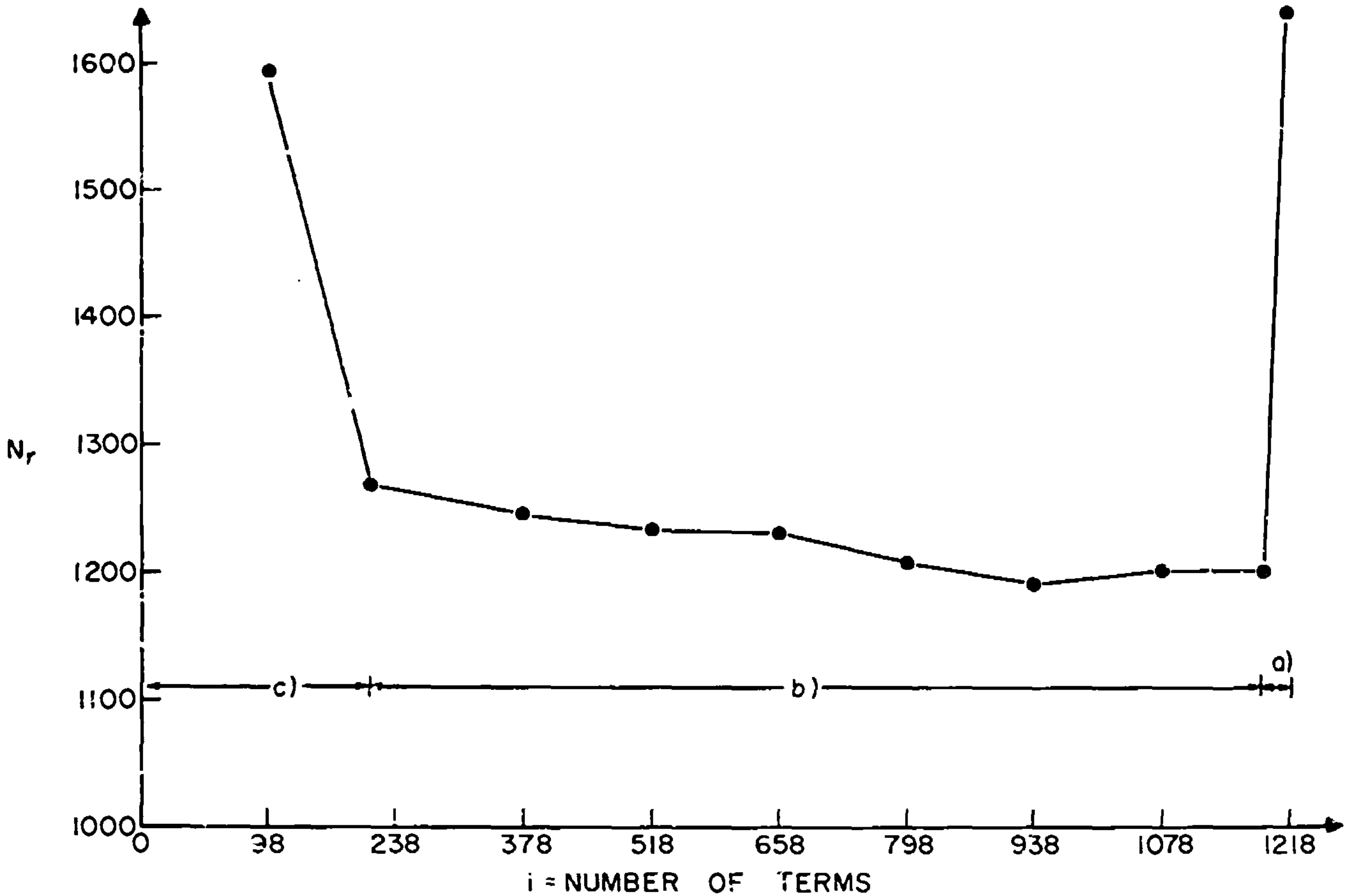
When the terms are deleted in <u>increasing Q_i order</u>, the Q_I and N_r curves come out very much as predicted (Figure 3 and 4), being both of approximately the same shape: dipping down to a minimum and shooting off at both ends (see Figure 5 for comparison). Interestingly, no documents are "lost" (have all their keywords deleted) until all but 98 keywords are deleted, at which time N_r shoots up, indicating that these 98 terms are real discriminators. Also, the N_r curve has a very large, flat middle "minimum" (discounting noise) area — deleting 28 or 36 x 28 terms does not make much difference.

The keywords are thus divided into 3 sets (Figure 4):



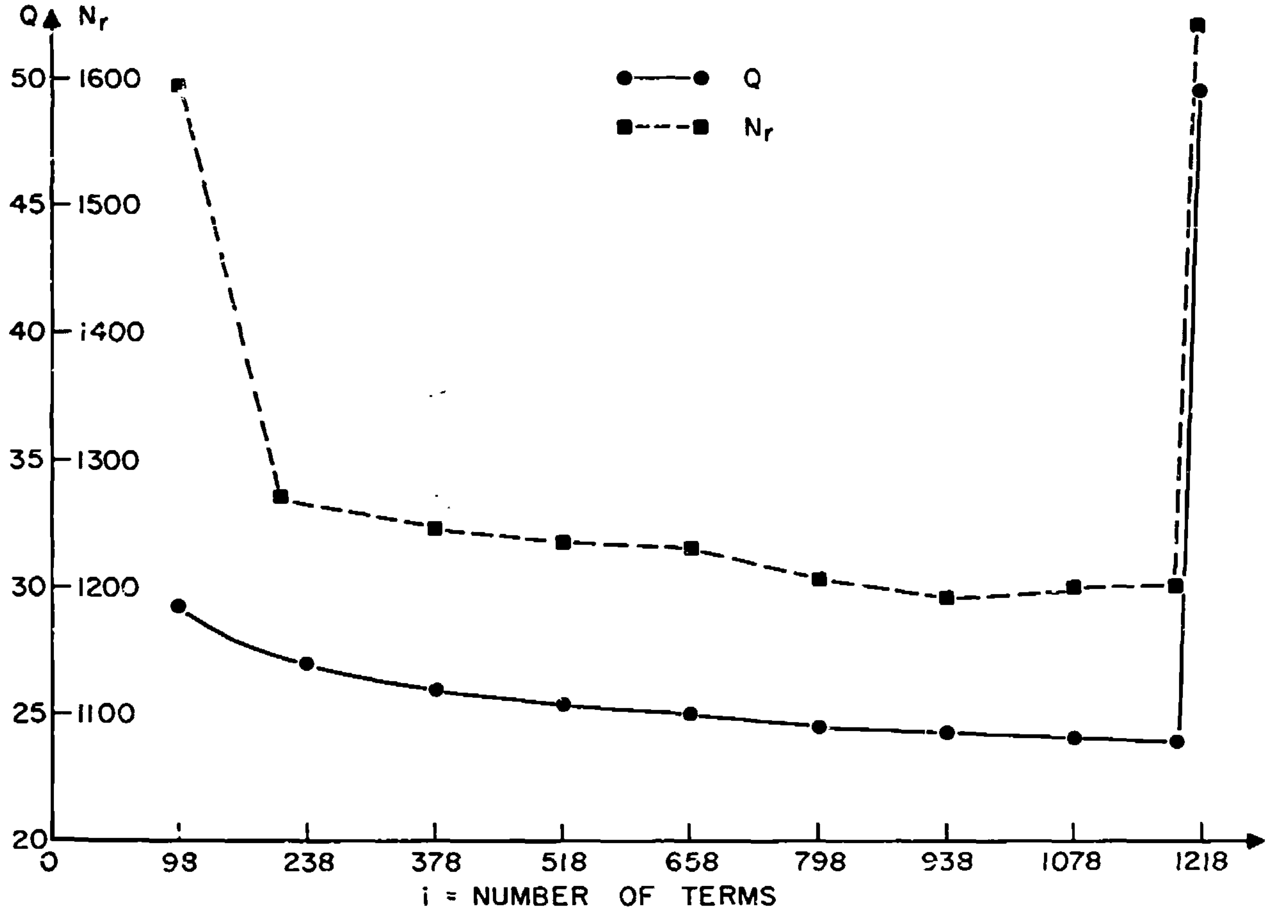
Q vs NUMBER OF CONCEPTS – DELETED BY Q ORDER

Figure 3



N_r (TOTAL FOR 33 QUERIES) vs NUMBER OF CONCEPTS—DELETION BY Q ORDER

Figure 4



Q vs NUMBER OF CONCEPTS

N_r vs NUMBER OF CONCEPTS – DELETION BY Q ORDER

Figure 5

a) those on the right end whose deletion leads to better retrieval (lower N_r);

b) the middle terms which do not make much difference;

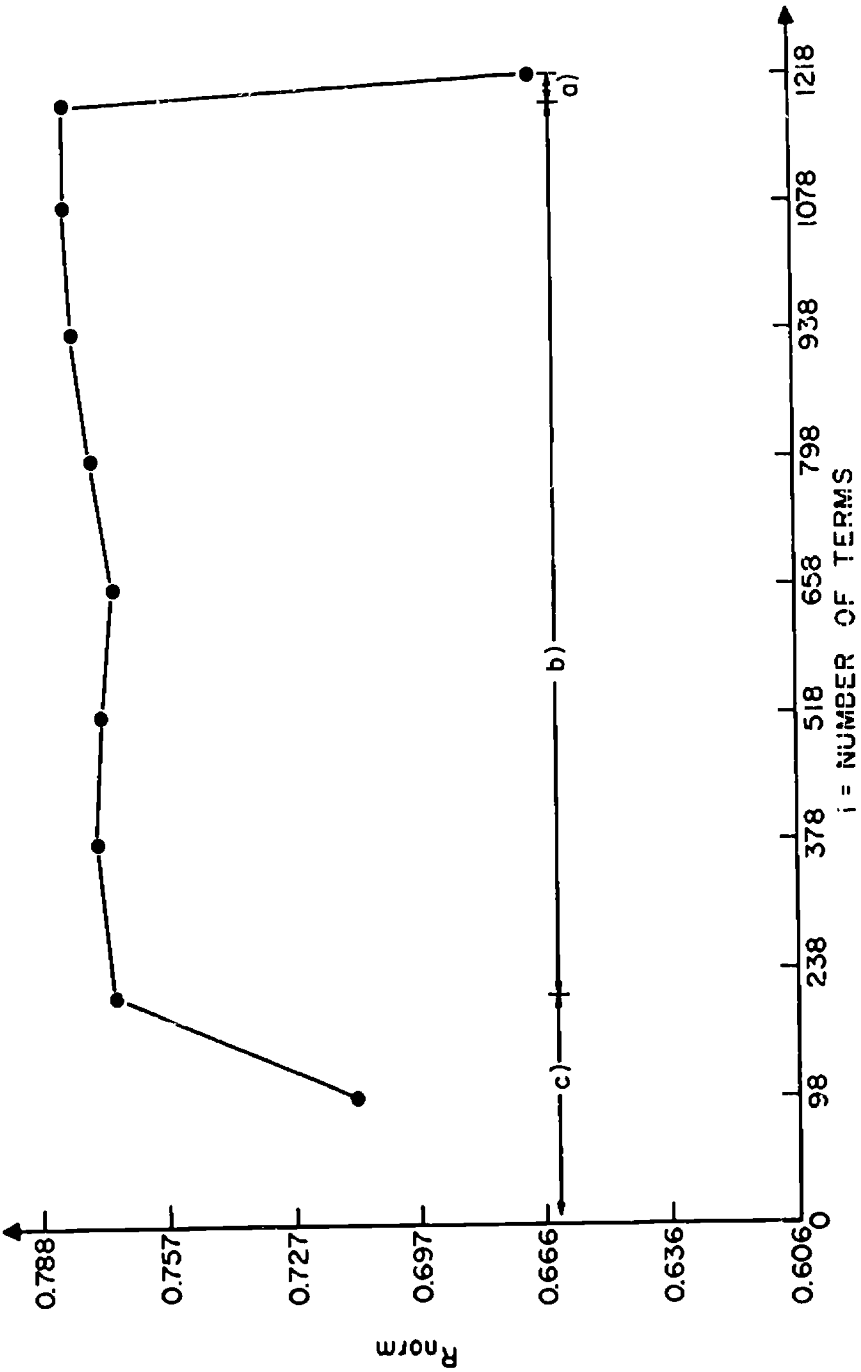
c) those at the left end which must be retained for good retrieval.

The sharp drop on the right-hand side of the curves is somewhat misleading. If all the points along the drop were plotted (corresponding to deleting 1, 2, 3, . . ., 28 keywords), it could be seen that the minimum actually occurs after the first 10 terms are deleted. These 10 terms constitute the set a), and it turns out that for all 10 terms, $Q_i < Q$ (Q without subscript is Q for the full index language). That is, these terms are of the type which according to predictions could be dropped from the index language, and the N_r curve shows that they should be. For all other terms (sets b) and c)), $Q_i > Q$. The members of set a) are therefore easy to identify and include in a negative dictionary: calculate Q for the full index language and Q_i for each keyword and put in the negative dictionary those keywords with $Q_i < Q$.

The normalized recall, defined by

$$R_{norm} = 1 - \frac{\sum_{i=1}^{n} (r_i - i)}{n \cdot (N - n)}$$

for N the total number of documents, n the number of relevant documents and r_i the rank of the i^{th} relevant document retrieved, is an alternate measure of retrieval effectiveness. The curve of normalized recall vs. terms deleted (Figure 6) delineates the same sets a), b), and c) that the N_r curve did. Since high recall is an indication of good retrieval (as opposed to low N_r), ...ng the recall curve (by subtracting all values from 1) is required to



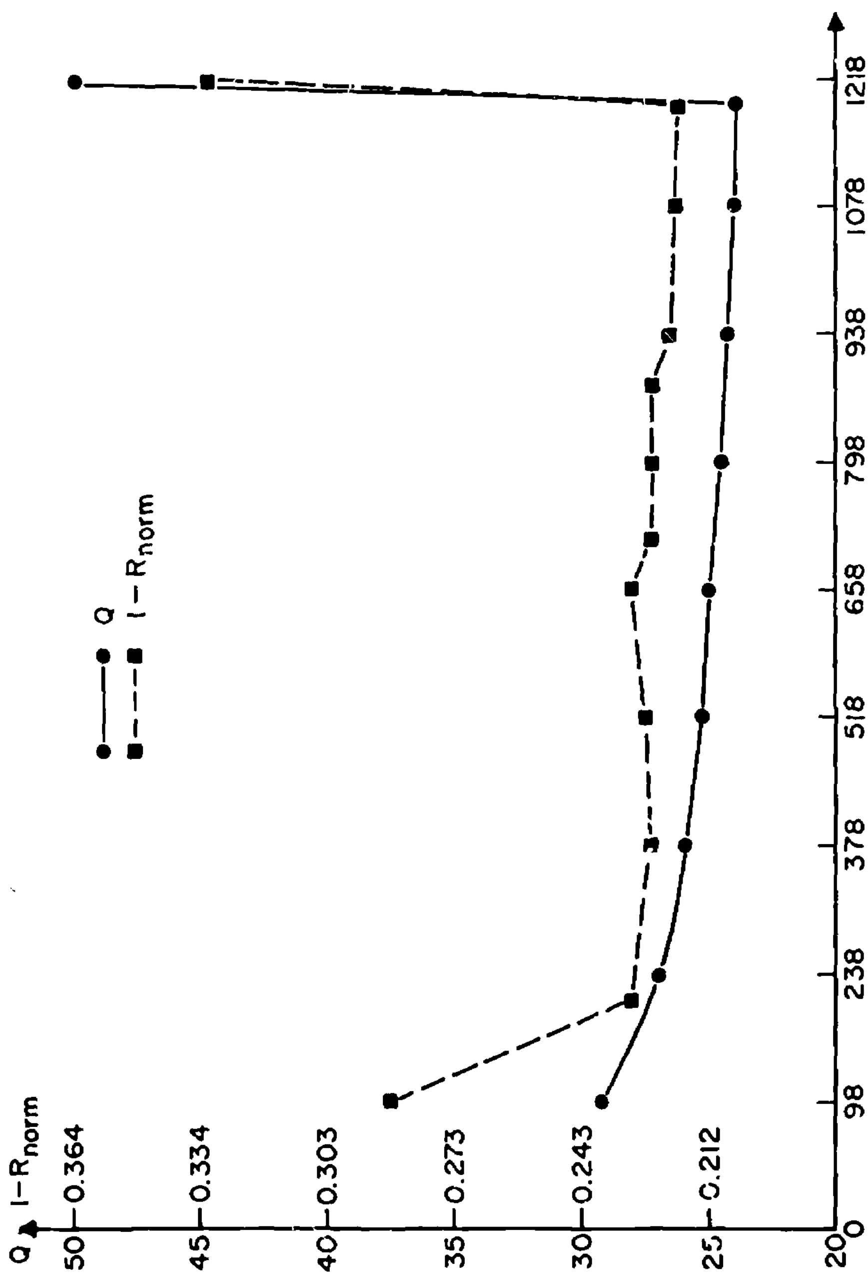
NORMALIZED RECALL vs NUMBER OF CONCEPTS-DELETION BY Q ORDER

Figure 6

show that recall also follows the pattern of Q (Figure 7).

It is interesting to note the frequency classes into which the sets a), b), and c) fall. The non-discriminating members of set a) exhibit the highest frequencies (40% - 100%); the "in-between" members of set b) have the lowest frequencies (0% - 10%), while the discriminators of set c) have 10% - 40%. While the terms in each set occur in the above ranges, within a set they are not exactly in frequency order. Therefore, in terms of frequency, the dividing line between discriminators and non-discriminators is not a clear one, and its absolute value (here, 40%) is likely to change from collection to collection. The use of relative Q's to separate out the non-discriminators, however, does not require the choice of such a cut-off point, and is an easier criterion to apply in constructing a negative dictionary.

When the terms are deleted in <u>decreasing frequency order</u>, the predicted curves do not show up (Figure 8 and 9). Q is strictly decreasing (reading from the right) — the more terms deleted, the more the space spreads out. Since the terms are dropped in approximately the order a), c), b), the loss of non-discriminator a) terms causes the same initial dip. Since the c) terms occur in more documents (have higher frequencies) than the b) terms, deleting them continues the process of spreading out the document space, until documents are identified only by a stray, "rare" word from set b). (In Q order, deleting terms from set b) has the opposite effect; documents that were "pulled away" from the centroid by odd words now move in closer together as terms from set b) are deleted, and Q goes up.) N_r has its initial dip resulting from the loss of the terms of set a), and then rises sharply as the discriminating terms of set c) are lost and the remaining keywords prove to be poor identifiers. In this case, documents

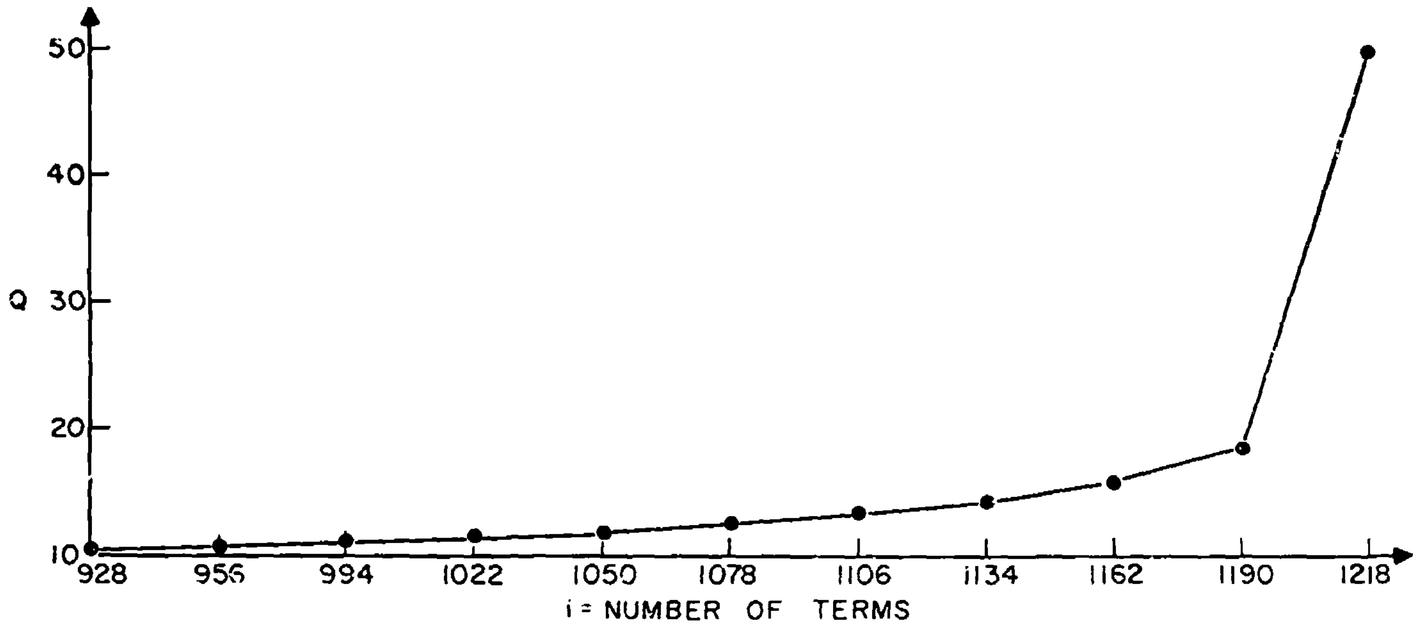


i = NUMBER OF TERMS

Q vs NUMBER OF CONCEPTS

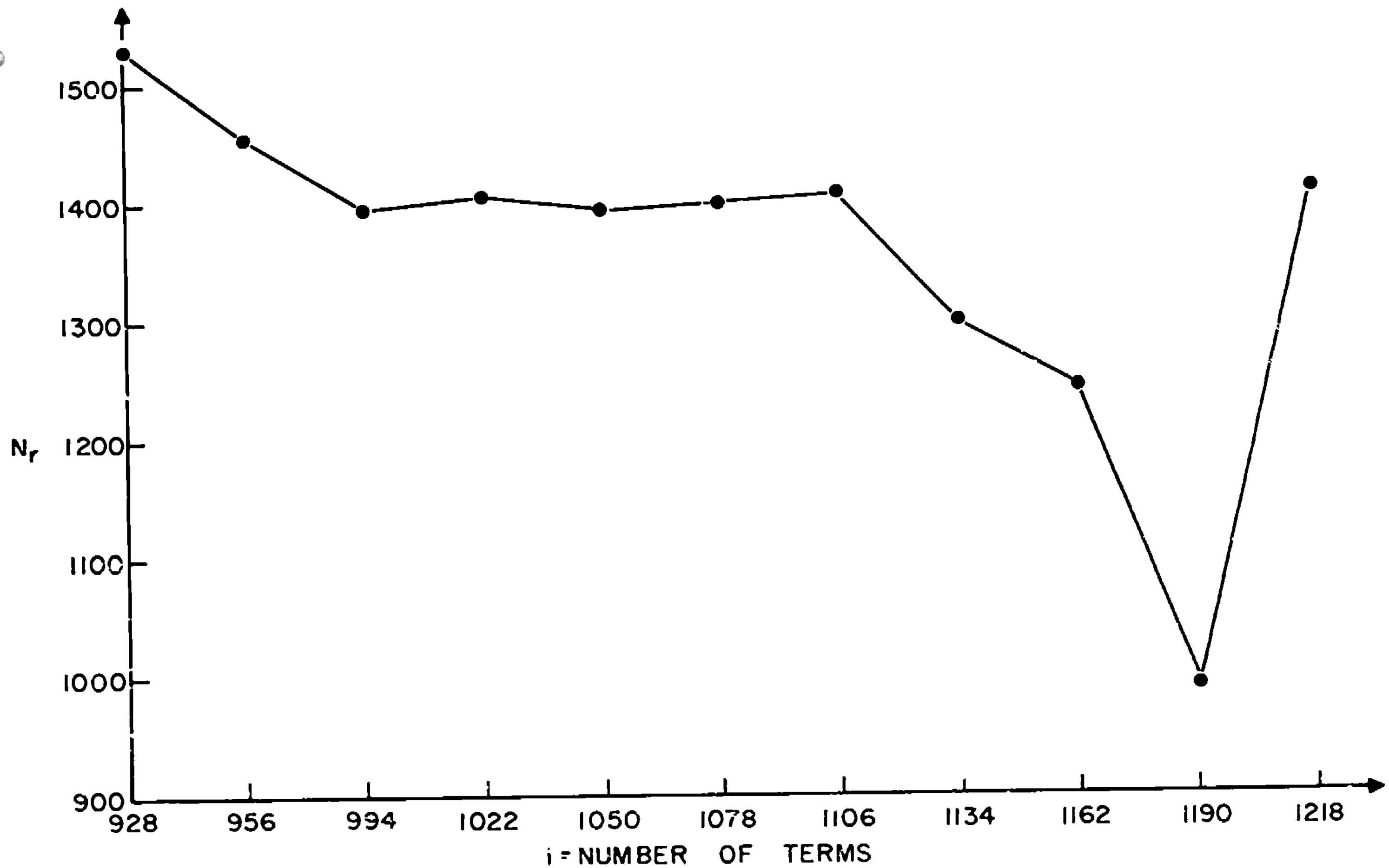
1-R_{norm} vs NUMBER OF CONCEPTS—DELETED BY Q ORDER

Figure 7



Q vs NUMBER OF CONCEPTS - DELETED BY FREQUENCY

Figure 8



N_r (TOTAL FOR 29 QUERIES) vs NUMBER OF CONCEPTS—DELETED BY FREQUENCY

Figure 9

are "lost" much more quickly, after only 560 keywords are deleted.

It is interesting to look at the keywords that fall into sets a), b), and c). Table 1 gives the 10 members of set a) in increasing Q order and their frequencies of occurrence (out of 82).

Keyword	Frequency
off	78
the	77
and	80
a	62
in	61
for	54
to	53
information	44
is	46
are	38

Table 1

Nine of the ten are identifiable as "common function words" without particular semantic content. The tenth, the term "information", also shows up as a non-discriminator, for this particular collection. Since the ADI collection covers documentation, this is not surprising. The fact that "information" does occur in set a) is an indication that the Q criterion will be helpful in constructing negative dictionaries tailored to the collection with which they will be used.

When 40 x 28 terms are deleted, the 98 which remain comprise set c), the so-called discriminators. Many of the 98 can classify as "content words" — "request", "education", "thesaurus", "retrieve" (see Table 2). On the other hand, several "function words" also occur, e.g., "at", "as", "it", "not", "has", "was". That is, in the ADI collection composed of abstracts (rather than full texts), these words serve to "distinguish" between those

Keyword	Frequency	Keyword	Frequency	Keyword	Frequency
index	19	usage	12	tape	7
library	10	procedure	7	produce	11
science	12	national	6	role	8
exchange	3	chemical	5	manual	6
search	12	program	17	recognition	3
process	14	publication	5	editing	2
service	10	journal	10	new	11
documents	19	logic	4	been	13
center	7	reference	6	not	4
definition	3	as	23	rules	2
technical	9	mechanized	3	remote	1
computer	23	it	9	interrogation	1
read	6	communication	7	microfilm	4
character	5	test	5	has	15
copy	7	can	11	prepare	5
be	16	education	4	graduate	3
book	3	material	4	into	5
use	13	by	27	an	27
at	18	concept	7	training	6
retrieve	28	need	11	that	11
analysis	7	level	3	abstract	5
file	6	organization	7	catalogue	1
date	14	facet	1	mathematical	1
thesaurus	4	vocabulary	4	access	5
system	33	have	10	store	7
from	17	or	15	handle	8
method	13	which	14	school	4
page	5	citation	4	literature	5
transfromation	2	comparison	4	word	5
machine	11	relation	5	was	5
image	1	request	5	IBM	4
text	7	foreign	1	name	2
automatic	8	special	8		

Keywords are in decreasing Q_i order, reading down the columns. That That is, "index" is the best discriminator, bein better than "technical", which is better than "usage", which is better than "tape", which is better than "name", which is the worst discriminator in set c).

Set c) -- Discriminators

Table 2

"documents" in which they appear and those in which they do not. Again, the Q criterion is matching the dictionary to the collection to produce maximal retrieval in a mechanical way without the benefit of human judgment.

The members of set b) appear in an average of two documents each. Both "function words" like "would" and "content words" like "overdue" and "efficiency" are found. Since function words are found in all three sets (and therefore at all frequency ranges), it is clear that a criterion of frequency of occurrence alone is not going to find all function words. At the same time, it will not be a good judge of true discriminators.

4. Experimental Method

The above results are produced in an three-step process:

1) a LOOKUP run produces full document and query vectors, and a list of all word stems used;

2) a FORTRAN program reads document-term vectors, calculates Q_i for each term i and produces a file in <u>increasing Q_i order</u> of keyword concept numbers, frequency of occurrence, and their total sum of weights (over all documents). A second program sorts this file into <u>decreasing frequency order</u>;

3) a third program works with the full documents and query vectors, and either of the term-frequency-weight files to perform the deletion of keywords and the search runs.

A) Calculating Q_i

The first program inverts the document-term vectors and works with this new file and the term-frequency-weight file it creates. It finds the elements of the centroid vector $\underline{c}$ by dividing the total sums of weights for each term by N, the number of documents. To calculate Q, it saves $\sum_{i=1}^{t} v_{ij}^2$ for each document j, and $\sum c_i^2$ for the centroid. Then

$$Q = \sum_{j=1}^{N} \frac{\sum_{i=1}^{t} v_{ij} \cdot c_i}{\sqrt{\sum v_{ij}^2 \cdot \sum c_i^2}} = \frac{1}{\sqrt{\sum c_i^2}} \sum_{j=1}^{N} \frac{\sum_{i=1}^{t} v_{ij} \cdot c_i}{\sqrt{\sum v_{ij}^2}}$$

where t is the total number of terms, and the values of v_{ij} are obtained from the term-document file. As the program goes along, it also saves $\sum_{i=1}^{t} v_{ij} \cdot c_i$ for each document j. Then

$$Q_k = \frac{1}{\sqrt{\sum^{t}(c_i^2) - c_k^2}} \sum_{j=1}^{N} \frac{\sum_{i=1}^{t}(v_{ij} \cdot c_i) - v_{kj} \cdot c_k}{\sqrt{\sum^{t}(v_{ij}^2) - v_{kj}^2}}$$

where the sums to t are all stored values and the values involving k are in the program's files.

B) Deleting and Searching

The third program also inverts the document-term file, and keeps track of $\sum v_{ij}^2$ for all documents j, adjusting the values of the sums as terms are deleted. This program finds $\sum c_i^2$ and calculates $Q_{\{1-28\}}$, $Q_{\{1-56\}}$, . . . , in a manner similar to that described above.

To perform searching a query $\underline{w}$ and its relevancy decisions are read in. Using pointers to keep track of which terms are deleted (which part of the term-document file to ignore), the query is correlated with each document in the collection of full vectors, then with document vectors with 28 terms deleted, then with 56 deleted, and so on. The cosine $\sum v_{ij} \cdot w_i$ /

$\sqrt{\sum v_{ij}^2 \cdot \sum w_i^2}$ can be calculated, since the $\sum v_{ij}^2$ are stored, the v_{ij} are in the inverted term-document file, and w was just read in. The ranks of the relevant documents can be found by comparing cosines (number of documents with a higher cosine = rank - 1). Typical results are shown in Table 3. The output format is as follows:

> the iteration number indicates how many groups of 28 keywords were deleted;
>
> C1 = average cosine of the relevant documents;
>
> C2 = normalized recall;
>
> N_r = rank of last relevant document;
>
> Q = Q_I for the iteration given by the iteration number;
>
> nR → document n is relevant; the next two numbers are its rank and correlation with the query.

The SMART routine AVERAGE is used to compare retrieval results for different index languages. Some of the results for deleting terms in increasing Q_i order, in particular, iterations 0, 1, 9, 36, and 40, are shown in Figure 10 (which labels these Run 0, 1, 2, 3, and 4, respectively). The recall-precision curves show that deleting concepts does improve retrieval effectiveness. By comparing entries in the table of recall-precision values (Table 4), it can be seen that Run 1 falls on top of Run 2. That is, retrieval performance is about the same whether 28 or 9 x 28 keywords are deleted, but in either case, performance is better than when no terms are deleted. And when only 98 keywords are left (Run 4), the performance is still better than with the full index language (Run 0), falling halfway between best and worst.

To test the effectiveness of the negative dictionary created by the

Iteration 5 Query 24
C1=0.196 C2=0.9710807
NR 22 Q 23.997940

3R	1	0.3816933
72R	2	0.2828426
21R	3	0.2480695
59R	4	0.2422719
45R	6	0.1556997
10R	7	0.1490711
76R	9	0.1204828
43R	10	0.1195228
14R	22	0.0609837

Iteration 6 Query 24
C1=0.196 C2=0.9710807
NR 22 Q 24.046610

3R	1	0.3816933
72R	2	0.2828426
21R	3	0.2480695
59R	4	0.2422719
45R	6	0.1556997
10R	7	0.1490711
76R	9	0.1204828
43R	10	0.1195228
14R	22	0.0609837

Iteration 7 Query 24
C1=0.198 C2=0.9710807
NR 22 Q 24.113150

3R	1	0.3816933
72R	2	0.2828426
21R	3	0.2666666
59R	4	0.2458614
45R	6	0.1556997
10R	7	0.1490711
76R	9	0.1204828
43R	10	0.1195228
14R	22	0.0609837

Iteration 8 Query 24
C1=0.199 C2=0.9710807
NR 22 Q 24.160200

3R	1	0.3816933
72R	2	0.2828426
21R	3	0.2666666
59R	4	0.2535462
45R	6	0.1556997
10R	7	0.1490711
76R	9	0.1204828
43R	10	0.1195228
14R	22	0.0612373

Iteration 0 Query 25
C1=0.426 C2=0.7594937
NR 54 Q 49.449810

53R	1	0.5960834
13R	8	0.4109974
24R	54	0.2695820

Iteration 1 Query 25
C1=0.221 C2=0.8101266
NR 48 Q 23.936350

13R	1	0.3608438
53R	2	0.3015113
24R	48	0.0000000

Iteration 2 Query 25
C1=0.221 C2=0.8101266
NR 48 Q 23.936350

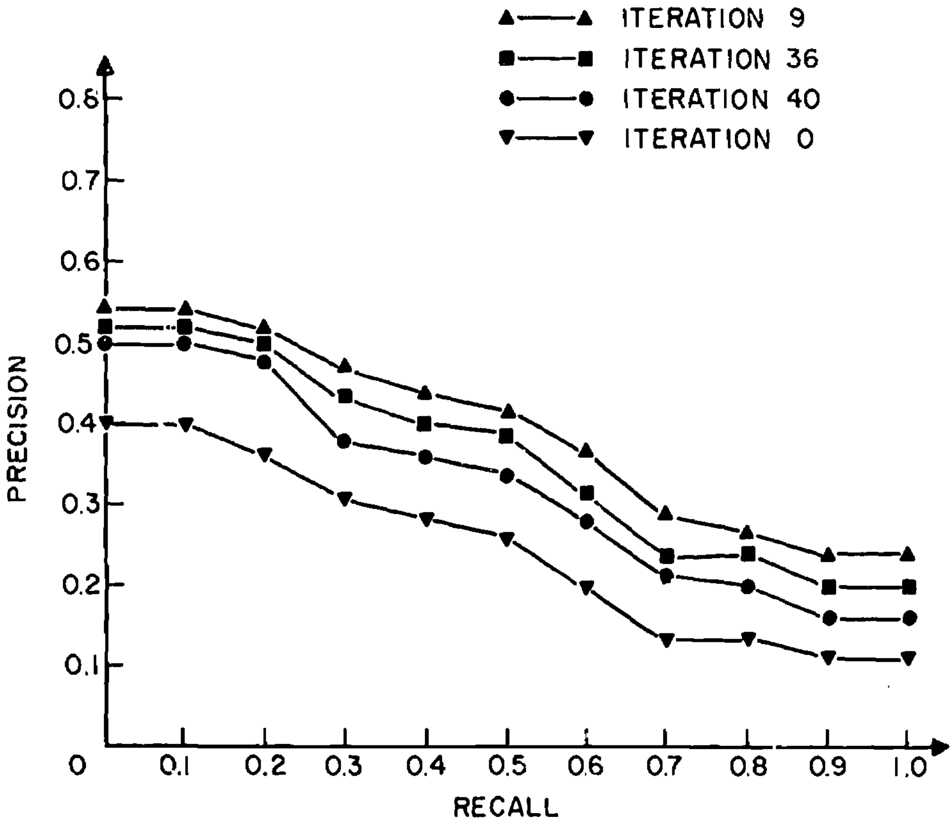
13R	1	0.3608438
53R	2	0.3015113
24R	48	0.0000000

Iteration 3 Query 25
C1=0.221 C2=0.8101266
NR 48 Q 23.937240

13R	1	0.3608438
53R	2	0.3015113
24R	48	0.0000000

Typical Output

Table 3



RECALL LEVEL AVERAGES

Figure 10

Run 0 — 33 Queries (Plus 0 Nulls) — ADI-82 Full
Run 1 — 33 Queries (Plus 0 Nulls) — ADI-82 Minus 28
Run 2 — 33 Queries (Plus 0 Nulls) — ADI-82 Minus 252
Run 3 — 33 Queries (Plus 0 Nulls) — ADI-82 Minus 1092
Run 4 — 33 Queries (Plus 0 Nulls) — ADI-82 Minus 1120

	Run 0		Run 1		Run 2		Run 3		Run 4	
Recall	NQ	Precision	NQ	Precision	NQ	Precision	NQ	Precision	NQ	Precision
0.0	0	0.4027	0	0.5367	0	0.5498	0	0.5271	0	0.5069
0.05	1	0.3951	1	0.5367	1	0.5405	1	0.5271	1	0.5069
0.10	2	0.3926	2	0.5367	2	0.5405	2	0.5271	2	0.4976
0.15	4	0.3926	4	0.5177	4	0.5215	4	0.5098	4	0.4976
0.20	11	0.3638	11	0.5122	11	0.5159	11	0.4991	11	0.4769
0.25	16	0.3537	16	0.4787	16	0.4819	16	0.4501	16	0.3948
0.30	16	0.3277	16	0.4637	16	0.4623	16	0.4464	16	0.3851
0.35	22	0.2749	22	0.4439	22	0.4454	22	0.4035	22	0.3558
0.40	22	0.2740	22	0.4439	22	0.4454	22	0.4035	22	0.3551
0.45	22	0.2615	22	0.4264	22	0.4280	22	0.3832	22	0.3424
0.50	29	0.2612	29	0.4262	29	0.4277	29	0.3821	29	0.3424
0.55	29	0.2037	29	0.3662	29	0.3669	29	0.3165	29	0.2864
0.60	29	0.2031	29	0.3647	29	0.3658	29	0.3141	29	0.2823
0.65	29	0.2025	29	0.3622	29	0.3603	29	0.3138	29	0.2658
0.70	29	0.1468	29	0.2798	29	0.2748	29	0.2490	29	0.2128
0.75	29	0.1461	29	0.2798	29	0.2748	29	0.2490	29	0.2128
0.80	29	0.1390	29	0.2673	29	0.2671	29	0.2385	29	0.1947
0.85	29	0.1294	29	0.2466	29	0.2529	29	0.2232	29	0.1815
0.90	29	0.1194	29	0.2317	29	0.2391	29	0.2090	29	0.1588
0.95	29	0.1178	29	0.2317	29	0.2391	29	0.2090	29	0.1588
1.00	33	0.1178	33	0.2305	33	0.2379	33	0.2078	33	0.1576
Norm Recall		0.6687		0.7798		0.7789		0.7692		0.7182
Norm Precision		0.4490		0.5750		0.5754		0.5585		0.5084
Rank Recall		0.1459		0.2743		0.2783		0.2498		0.1943
Log Precision		0.2829		0.4043		0.4070		0.3665		0.3387

NQ = Number of Queries used in the average not dependent on any extrapolation.
Norm = Normalized

Recall — Level Averages

Table 4

Q criterion (i.e., the dictionary consists of the terms in set a)), retrieval results should be compared with those obtained on the same collection using the 204 "common English words" list as a negative dictionary. The latter collection is not available on the SMART system, so results are compared with those obtained using the thesaurus dictionary, which lumps synonyms together as well as deleting the 204 words. As shown in Figure 11, the results with the Q negative dictionary (Run 1 = iteration 1) are just about the same as those for the thesaurus, except in the low recall area. Since thesaurus construction involves a large amount of hand work and human judgment while the Q negative dictionary can be generated mechanically, the Q method is preferable if high recall is desired, and the time and effort saved by not preparing a thesaurus may justify the use of the Q method even if precision is the goal.

5. Cost Analysis

The basic rationale for negative dictionaries is that they delete many of the frequent keywords, thus reducing the size of files, and lowering storage and search costs. There is a tradeoff between file size and retrieval effectiveness, and a point of balance between the two has to be found. From Figure 10, it can be deduced that deleting 9 x 28 terms leads to about the same retrieval results as deleting only 28 terms, and if any terms are dropped, all 252 can be. However, deleting 36 x 28 (Run 3) lowers retrieval performance only slightly. Is the saving worth deleting the extra terms?

The question can be rephrased as follows: what is the saving in costs when extra terms from set b) are deleted? The keywords in set a) are deleted to improve retrieval (Figure 10, Run 1). Deletion of keywords in set b) has a lesser effect on retrieval (Run 2 and 3), but the terms in

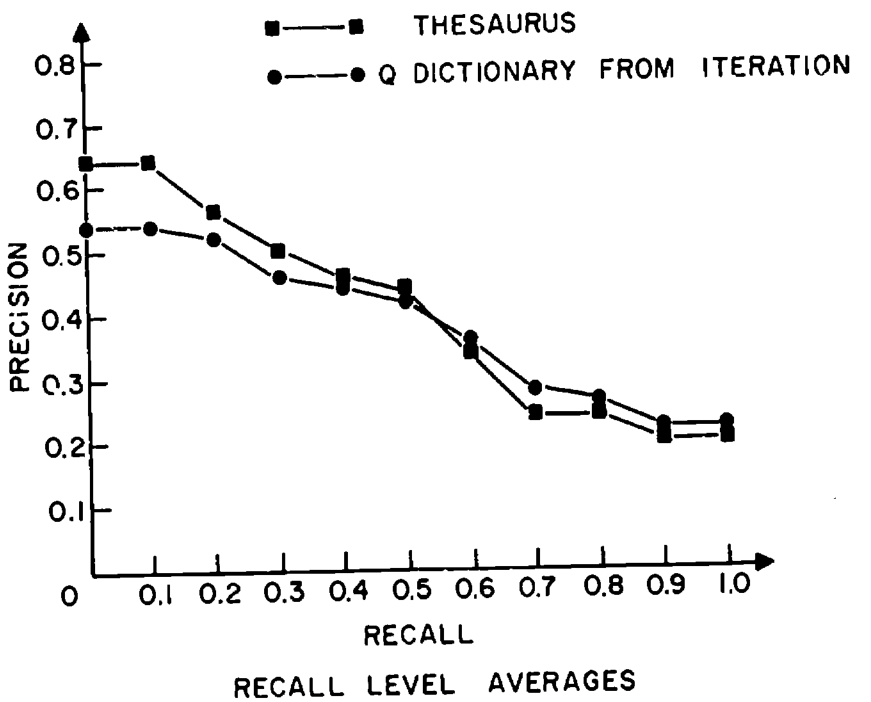

THESAURUS
Q DICTIONARY FROM ITERATION I
PRECISION
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
RECALL
RECALL LEVEL AVERAGES

Figure 11

set b) constitute the bulk of the terms to be stored. How much do they cost versus how much do they add to retrieval?

The cost accounting will depend on the system being used and the kind of results it produces. Assume a print-out of all retrieval documents is required and the system works as follows:

a) a full search is performed for each query, processed separately;

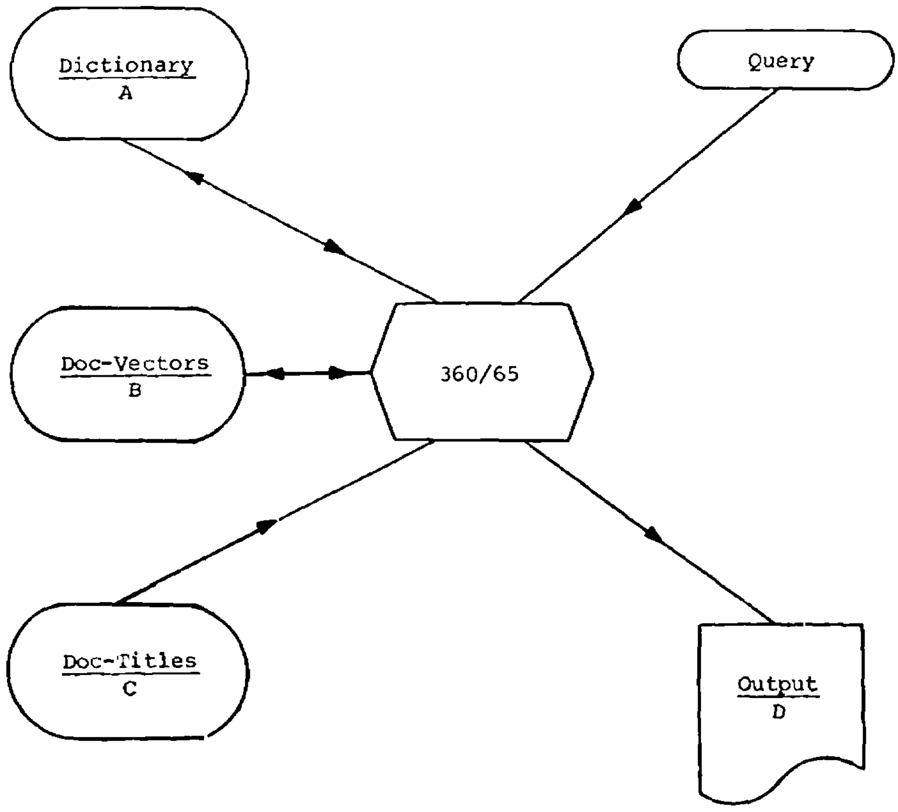
b) results are in the form "Document Title" and "Reference Number", one line per document, with all documents retrieved printed out;

c) the computer is the 360/65 under CLASP;

d) the search program uses 250K and the file organization of the SMART system.

Diagrammatically, the process will appear as in Figure 12. Queries are read in, one at a time, and looked up in the dictionary (A). Each query is correlated with all members of the document file (B) and ranked. The document titles for all documents up to the last relevant are found in the title file (C) and returned to the user (D). (Using all documents up to the last relevant is a convenient measure of how many documents the average user will see.)

What is the dependence of these operations on the total number of terms t? Step (A) is independent of t — each word of the query must be checked for occurrence in the dictionary; non-occurrence takes as long to discover as occurrence. The search step (B) depends on t in two ways: as general file size is reduced, accessing time will go down, and as vector length is reduced, the number of calculations required to compute query-document correlations will be lower. Steps (C) and (D) are independent of t, but are a function of N_r, the rank of the last relevant document



System Organization

Figure 12

(since all documents with rank $\leq N_r$ are printed, relevant or not).

Accessing time is related to number of disc tracks read. The ADI collection with all keywords included occupies 4 tracks. Deleting about 200 terms will reduce the number to 3, but even if all the terms found in set b) are deleted, the number of tracks required remains at 3. For 35 queries, the total time saved with reduction to 3 tracks is 1.2 sec. In addition, 50 millisec. is saved in computation time, or for 200 terms deleted, 10 more sec. saved.

The rank of the last relevant document, N_r, generally increases as terms are deleted, resulting in more output lines and an increase in time and cost. Table 5 gives exact figures, in terms of dollars saved, when various numbers of terms are deleted. Figure 13 is a plot of these values, showing the savings in search resulting from deduction from 4 tracks to 3, and the total savings, as functions of the number of terms deleted.

6. Conclusions

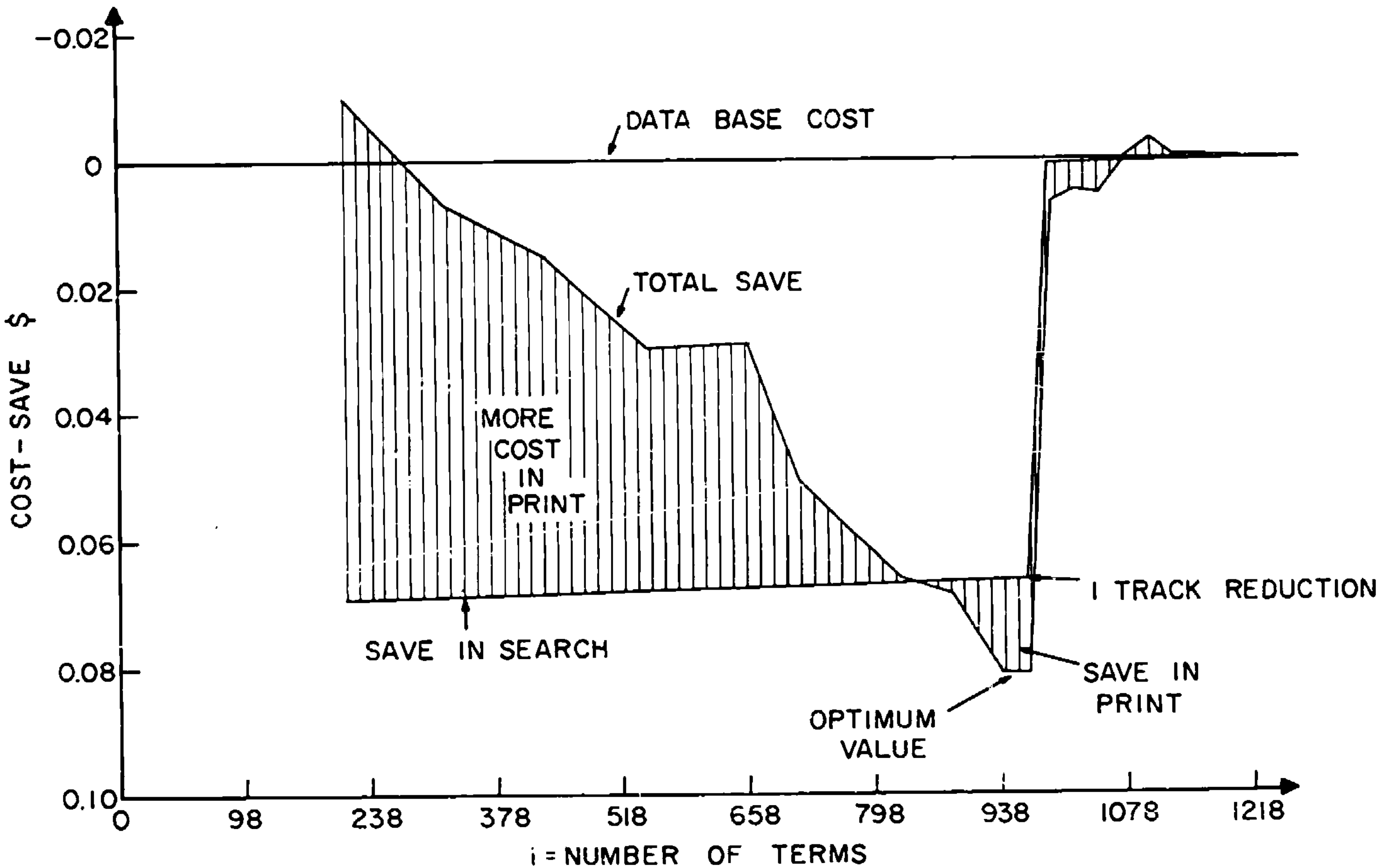
Clearly, a negative dictionary is needed; deletion of some keywords definitely improves retrieval. Deleting words in order of increasing Q seems the better method; while the N_r curve for frequency order has a lower minimum point, it is very unstable. Terms from set a), with $Q_i < Q$, are to be deleted; discriminators from set c) are to be retained. The question of what to do with the middle (set b)) depends on the needs of the user. For a large collection, deleting all but the most vital terms will save storage costs and search time, possibly at some small loss in retrieval. The ADI collection is too small to show very significant differences in cost when terms are deleted.

Number of terms remaining	Number of terms deleted from set b)	Save in Search (dollars)	Decrease in N_r (lines saved)	Save in Print (dollars)	Total Saved (dollars)
1190	0	0.0	0	0.0	0.0
1162	28	0.0	0	0.0	0.0
1134	56	0.00016	0	0.0	0.00016
1106	84	0.00024	- 2	-0.0026	-0.00236
1078	112	0.00033	0	0.0	0.00033
1050	140	0.00042	4	0.0052	0.00562
1022	168	0.0005	3	0.0039	0.0044
994	196	0.0006	5	0.0065	0.0071
966	224	0.0667	11	0.0143	0.0810
938	252	0.0668	11	0.0143	0.0811
882	308	0.0670	1	0.0013	0.0683
826	364	0.0671	- 1	-0.0013	0.0658
770	420	0.0672	- 6	-0.0078	0.0594
714	476	0.0674	-13	-0.0169	0.0505
658	532	0.0676	-29	-0.0377	0.0299
546	644	0.0678	-29	-0.0377	0.0301
434	756	0.0682	-41	-0.0533	0.0149
322	868	0.0685	-47	-0.0611	0.0074
210	980	0.0688	-61	-0.0793	-0.0105

In terms of cost, the optimal number of terms to delete from set b) is about 950.

Cost Statistics

Table 5



COST - SAVE vs NUMBER OF CONCEPTS - REGION "B" DELETED BY Q

Figure 13

The algorithm presented for determing the set a) requires the calculation of Q_i for each term i, and the storage of the entire term-document file. By judicious handling of the values involved, a farily efficient method for discovering set a) is produced. This procedure should be reasonably practical to run on a large collection, at least for generating the initial negative dictionary. Updates for the dictionary when the collection changes could be produced by rerunning the programs on a representative sample of the revised collection.

References

[1] G. Salton, Automatic Information Organization and Retrieval, McGraw-Hill Series in Computer Science, 1968.

[2] G. Salton and M. E. Lesk, Information Analysis and Dictionary Construction, Information Storage and Retrieval, Report No. ISR-11 to the National Science Foundation, Section IV, Department of Computer Science, Cornell University, June 1966.

[3] E. M. Keen, Test Environment, Information Storage and Retrieval, Report No. ISR-13 to the National Science Foundation, Section I, Department of Computer Science, Cornell University, December 1967.

VII. Experiments in Automatic Thesaurus Construction for Information Retrieval

G. Salton

Abstract

One of the principal intellectual as well as economic problems in automatic text analysis is the requirement for language analysis tools able to transform variable text inputs into standardized, analyzed formats. Normally, word lists and dictionaries are constructed manually at great expense in time and effort to be used in identifying relationships between words and in distinguishing important "content" words from "common" words to be discarded.

Several new methods for automatic, or semi-automatic, dictionary construction are described, including procedures for the automatic identification of common words, and novel automatic word grouping methods. The resulting dictionaries are evaluated in an information retrieval environment. It appears that in addition to the obvious economic advantages, several of the automatic analysis tools offer improvements in retrieval effectiveness over the standard, manual methods in general use.

1. Manual Dictionary Construction

Most information retrieval and text processing systems include as a principal component a language analysis system designed to determine the "content", or "meaning" of a given information item. In a conventional library system, this analysis may be performed by a human agent, using established classification schedules to determine what content identifiers will best fit a given item. Other "automatic indexing" systems are known in which the content identifiers are generated automatically from document and query texts.

Since the natural language contains irregularities governing both the syntactic and the semantic structures, a content analysis system must normalize the input texts by transforming the variable, possibly ambiguous, input structures into fixed, standardized content identifiers. Such a language normalization process is often based on dictionaries and word lists, which specify the allowable content identifiers, and give for each identifer appropriate definitions to regularize and control its use. In the automatic SMART document retrieval system, the following principal dictionary types are used as an example [1]:

a) a negative dictionary containing "common" terms whose use is proscribed for content analysis purposes;

b) a thesaurus, or synonym dictionary, specifying for each dictionary entry, one or more synonym categories, or concept classes;

c) a phrase dictionary identifying the most frequently used word or concept combinations;

d) a hierarchical arrangement of terms or concepts, similar in structure to a standard library classification schedule.

While well-constructed dictionaries are indispensable for a consistent assignment of content identifiers, or concepts, to information items, the task of building an effective dictionary is always difficult, particularly if the environment within which the dictionary operates is subject to change, or if the given subject area is relatively broad and nonhomogeneous. [2]

The following procedure summarizes the largely manual process normally used by the SMART system for the construction of negative dictionaries and thesauruses [3]:

a) a standard common word list is prepared consisting of function words to be excluded from the dictionary;

b) a keyword-in-context, or concordance listing is generated for a sample document collection in the area under consideration, giving for each word token the context, as well as the total occurrence frequency for each word;

c) the common word list is extended by adding new non-significant words taken from the concordance listing; in general, the words added to form the revised common word list are either very high frequency words providing little discrimination in the subject area under consideration, or very low frequency words which produce few matches between queries and documents;

d) a standard suffix list is prepared, consisting of the principal suffixes applicable to English language material;

e) an automatic suffix removel program is then used to reduce all remaining (noncommon) words to word stem form; the resulting word stem dictionary may be scanned (manually) in order to detect inadequacies in the stemming procedure;

f) the most frequent significant word stems are then selected to serve as "centers" of concept classes in the thesaurus under construction;

g) the word stem dictionary is scanned in alphabetical order, and medium-frequency word stems are either added to existing concept classes, or are used as "centers" of new concept classes;

h) the remaining, mostly low frequency, word stems are inserted as members of existing word classes;

i) the final thesaurus is manually checked for internal consistency, and printed out.

It has been found experimentally that thesauruses resulting from these processing steps operate most satisfactorily if ambiguous terms are entered only into those concept classes which are likely to be of interest in the subject area under consideration — for example, a term like "bat" need not be encoded to represent an animal if the document collection deals with sports and ball games. Furthermore, the scope of the resulting concept classes should be approximately comparable, in the sense that the total frequency of occurrence of the words in a given concept class should be about equal; high frequency terms must therefore remain in classes by themselves, while low frequency terms should be grouped so that total concept frequencies are equalized. [3] A typical thesaurus excerpt is shown in Table 1 in alphabetical, as well as in numerical, order by concept class number. (Class numbers above 32,000 designate "common" words.) [4]

A number of experiments have been carried out with the SMART system in order to compare the effectiveness in a retrieval environment of manually constructed thesauruses, providing synonym recognition, with that of simple word stem matches in which word stems extracted from documents are matched with those extracted from queries. In general, it is found that the thesaurus procedure which assigns content identifiers representing concept classes, rather than word stems, offers an improvement of about ten percent in precision for a given recall level, when the retrieval results are averaged over many search requests.

Alphabetic Order		Numeric Order	
Word or Word Stem	Concept Classes	Concept Class	Words or Word Stems
wide	438	344	obstacle
will	32032		target
wind	345,233	345	atmosphere
			meteorolog
winding	233		weather
wipe	403		wind
wire	232,105	346	aircraft
			airplane
wire-wound	001		bomber
			craft
			helicopter
			missile
			plane

Typical Thesaurus Excerpt

Table 1

A typical recall-precision output is shown in Fig. 1 for thesaurus and word stem analysis processes. For the left-hand graph (Fig. 1 (a)) full document texts were used in the analysis, whereas document abstracts were used to produce Fig. 1 (b).* [5]

In order to determine what thesaurus properties are particularly desirable from a performance viewpoint, it is of interest to consider briefly the main variables which control the thesaurus generation process [6]:

a) word stem generation

   i) type of suffixing procedure used — whether fully automatic or based on a pre-existing suffix dictionary;

   ii) extent of suffixing — whether based on individual word morphology alone, or also incorporating word context;

b) concept class generation

   i) degree of automation in deriving thesaurus classes;

   ii) average size of thesaurus classes;

   iii) homogeneity in size of thesaurus classes;

   iv) homogeneity in the frequency of occurrence of individual class members (within a thesaurus class);

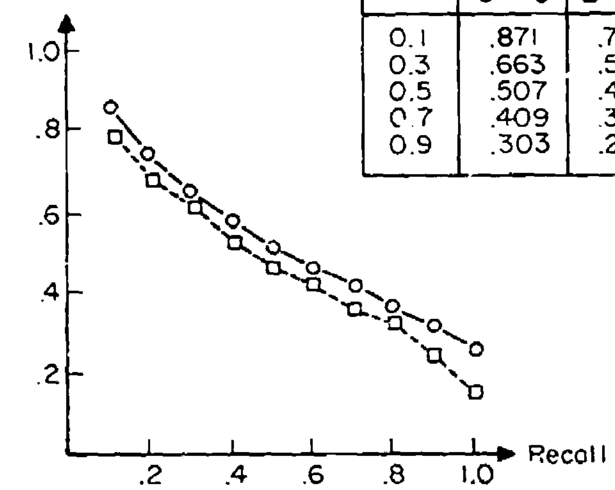
   v) degree of overlap between thesaurus classes (that is, number of word entries in common between classes);

   vi) semantic closeness between thesaurus classes;

---

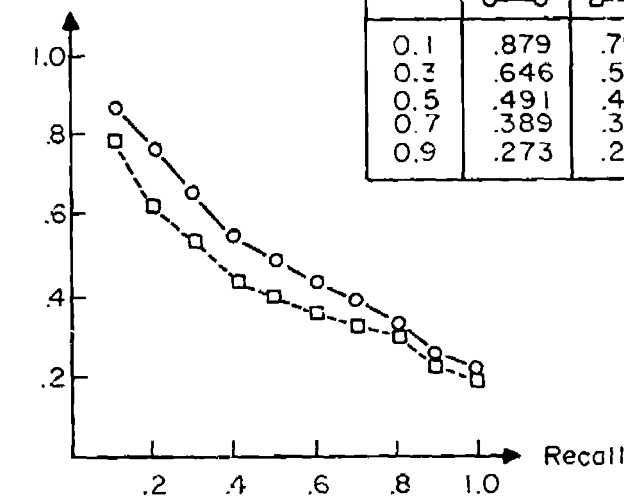
*Recall is the proportion of relevant material actually retrieved, while precision is the proportion of retrieved material actually relevant. In general, one would like to retrieve much of what is relevant, while rejecting much of what is extraneous, thereby producing high recall as well as high precision. The curve closest to the upper right-hand corner of a typical recall-precision graph represents the best performance, since recall as well as precision is maximized at that point.

o——o Manual Thesaurus
□- - -□ Word Stem Match



R	Precision o——o	Precision □- - -□
0.1	.871	.791
0.3	.663	.598
0.5	.507	.458
0.7	.409	.370
0.9	.303	.258

a) ADI Text



R	Precision o——o	Precision □- - -□
0.1	.879	.796
0.3	.646	.528
0.5	.491	.405
0.7	.389	.338
0.9	.273	.257

b) ADI Abstracts

Comparison of Manual Thesaurus and Word Stem Processes
(Averages over 82 documents, 35 queries)

Fig. 1

c) "common" word recognition

   i) degree of automation in common word recognition process;

   ii) proportion of common words as a percentage of the entire dictionary;

d) processing of linguistic ambiguities

   i) degree of automation in the recognition of linguistic ambiguities;

   ii) extent of recognition of ambiguous structures.

The language analysis procedures incorporated into the SMART document retrieval system all use an automatic word suffixing routine based on a hand-constructed suffix dictionary. Furtherm[illegible]tic ambiguities represented, for example, by the occurrence o[illegible]s in texts are not explicitly recognized by the SMART analy[illegible]ess.* The two main variables to be considered in examining these [illegible]tive-ness are therefore the common word recognition and the c[illegible]uping procedures. These two problems are treated in the remai[illegible]is study.

2. Common Word Recognition

In discussing the common word problem, it is imp[illegible] first of all, to distinguish common <u>function</u> words, such as prep[illegible], conjunc-

---

*Although several language analysis systems use elaborate procedures for the recognition of linguistic ambiguities [7,8], it appears that most potentially ambiguous structures are automatically resolved by restricting the application of a given dictionary to a specific, well-defined subject area.

tions, or articles, from common content words. The former are easily identified by constructing a list of such terms which may remain constant over many subject areas. The latter, typified by the word stem "automat" in a collection of computer science documents, consist of very high — or very low -- frequency terms which should not be incorporated into the standard concept classes of a thesaurus, because the respective terms do not adequately discriminate among the documents in the subject area under consideration. It is important that such words be recognized since their assignment as content identifiers would produce high similarity coefficients between information items which have little in common, and because their presence would magnify the storage and processing costs for the analyzed information items.

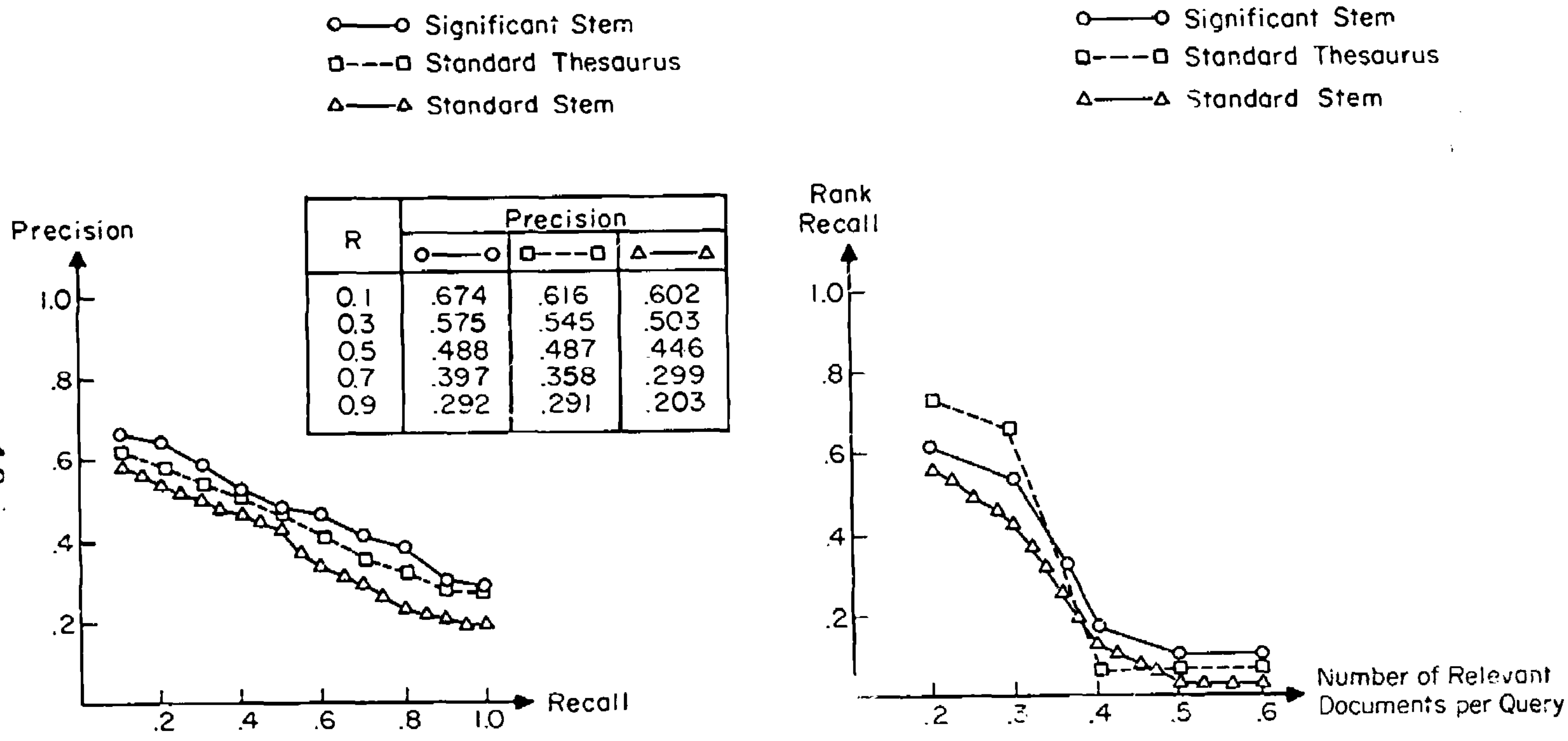
To determine the importance of the common content word recognition, a study was recently performed comparing the effectiveness in a retrieval environment of a standard word-stem matching process, a standard thesaurus, and a word-stem procedure in which the common content words normally identified as part of the thesaurus process were also recognized. [9] Specifically, a backward procedure was used to generate a word stem dictionary from a thesaurus by breaking down individual thesaurus classes and generating from each distinct word, or word stem, included in one of the thesaurus classes, an entry in the new stem dictionary. The main difference between this new significant stem dictionary and a standard stem dictionary is the absence from the dictionary of word stems corresponding to common functions and common content words normally identified only in a thesaurus. A comparison between significant and standard stem dictionaries will therefore produce evidence concerning the importance of common word deletion from document and query identifications, while the comparison between significant stem and thesaurus dictionaries leads to an evaluation of the concept classes and the term grouping methods used to generate the thesaurus.

A recall-precision graph for the performance of the three dictionary types is shown in Fig. 2(a), averaged over forty-two queries and two hundred documents in aerodynamics. It may be seen from Fig. 2(a) that the thesaurus produces an improvement of some ten percent in precision for a given recall value over the standard stem process. Unexpectedly, a further improvement is obtained for the significant stem dictionary over the thesaurus performance, indicating that the main virtue of the aerodynamics dictionary being tested is the identification of common words, rather than the grouping of term into concept classes. For the collection under study, the significant stem dictionary contains about twice as many common word entries as the standard stem dictionary.

Obviously, the recall-precision results reflected in the graph of Fig 2(a) cannot be used to conclude that synonym dictionaries, or thesauruses based on term grouping procedures are useless for the analysis of document and query content in information retrieval. Quite often, special requirements may exist for individual queries, such as, for example, an expressed need for very high recall, or precision; in such circumstances, a thesaurus may indeed turn out to be essential.

Consider as an example, the output graph of Fig. 2(b) in which a global evaluation measure, known as <u>rank recall</u>, is plotted for the ten queries (out of forty-two) which were identified by exactly six thesaurus concepts.* It is seen that for queries with very few relevant

*The rank recall measure expresses performance by a single number which [va]ries inversely with the ranks achieved by the relevant documents during [th]e retrieval process [1].



R	Precision o——o	Precision □- - -□	Precision △——△
0.1	.674	.616	.602
0.3	.575	.545	.503
0.5	.488	.487	.446
0.7	.397	.358	.299
0.9	.292	.291	.203

a) Recall–Precision Graph
(200 Documents, 42 Queries)

b) Rank Recall for Queries
with 6 Concepts

Comparison of Significant Stem Dictionary with Thesaurus
and Standard Stem (Cranfield Collection)

Fig. 2

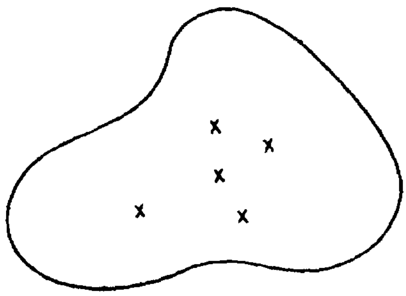
documents in the collection, the thesaurus in fact is able to identify the relevant items more effectively than either of the stem dictionaries. As the number of relevant documents per query increases, the stem methods catch up with the thesaurus process.

In view of the obvious importance of common word identification, one may inquire whether such entries might not be identifiable automatically, instead of being manually generated by the procedure outlined in the previous section. This question was studied using the following mathematical model. Consider the original set of terms, or concepts, used to identify a given query and document collection, and let this term set be altered by selective deletion of certain terms from the query and document identifications. One of two results will then be obtained depending on the type of terms actually removed:

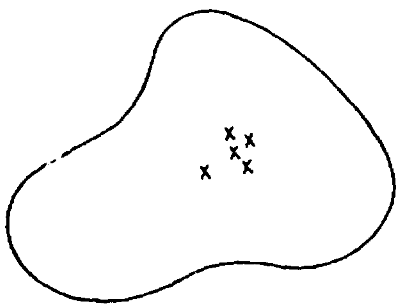
a) if the terms to be removed are useful for content analysis purposes, they will provide discrimination among the documents, and their removal will cause the document space to become more "bunched-up" by rendering all documents more similar to each other, that is, by increasing the correlation between pairs of documents;

b) on the other hand, if the terms being removed are common words which do not provide discrimination, the document space will spread out, and the correlation between document pairs will decrease.

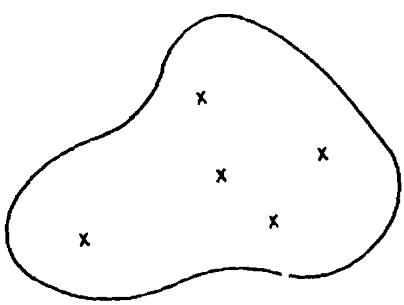
This situation is illustrated by the simplified model of Fig. 3, where each document is identified by 'x', and the similarity between two documents is assumed inversely proportional to the distance between corresponding x's. The conjecture to be tested is then the following: a term



a) Original Document Space



b) Document Space After Removal of Useful Discriminators



c) Document Space After Removal of Useless Nondiscriminators

Changes in Document Space Compactness Following Deletion of Certain Terms

Fig. 3

to be identified as a "common" word, and therefore to be removed from the set of potential content identifiers (and from the set of allowable thesaurus concepts) is one which causes the document space to spread out by decreasing its compactness.

The following procedure is used to verify the conjecture [10]. Consider a set of N documents, and let each document j be represented by a vector of terms, or concepts, $\underline{v}_j$, where $\underline{v}_{ij}$ represents the weight of term i in document j. Let the centroid $\underline{c}$ of all document points in a collection be defined as the "mean document", that is

$$\underline{c}_i = \frac{1}{N}\sum_{j=1}^{N} \underline{v}_{ij} ;$$

the centroid is then effectively the center of gravity of the document space. If the similarity, between pairs of documents i and j is given by the correlation $r(\underline{v}_i, \underline{v}_j)$, where r ranges from 1 for perfectly similar items to 0 for completely disjoint pairs, the compactness Q of the document space may be defined as

$$Q = \sum_{j=1}^{N} r(\underline{c}, \underline{v}_j), \qquad 0 \leq Q \leq N$$

that is, as the sum of the similarities between each document and the centroid; greater values of Q indicate greater compactness of the document space.

Consider then the function Q_i defining the compactness of the document space with term i deleted. If $Q_i > Q$, the document space is more compact and term i is a discriminator; contrariwise, if $Q_i < Q$, the space is more spread out, and deletion of term i may produce better retrieval. Since deletion of discriminators raises Q, and deletion of nondiscriminators (common words) lowers Q, an optimal set I of terms must exist such that Q_I becomes minimal.

The following experimental procedure may then be used:

a) consider each term i in order and compute Q_i;

b) arrange the terms in order of decreasing Q_i (that is, with terms causing the greatest decrease coming first);

c) define the set I of common terms to be deleted as the set leading to a minimal Q.

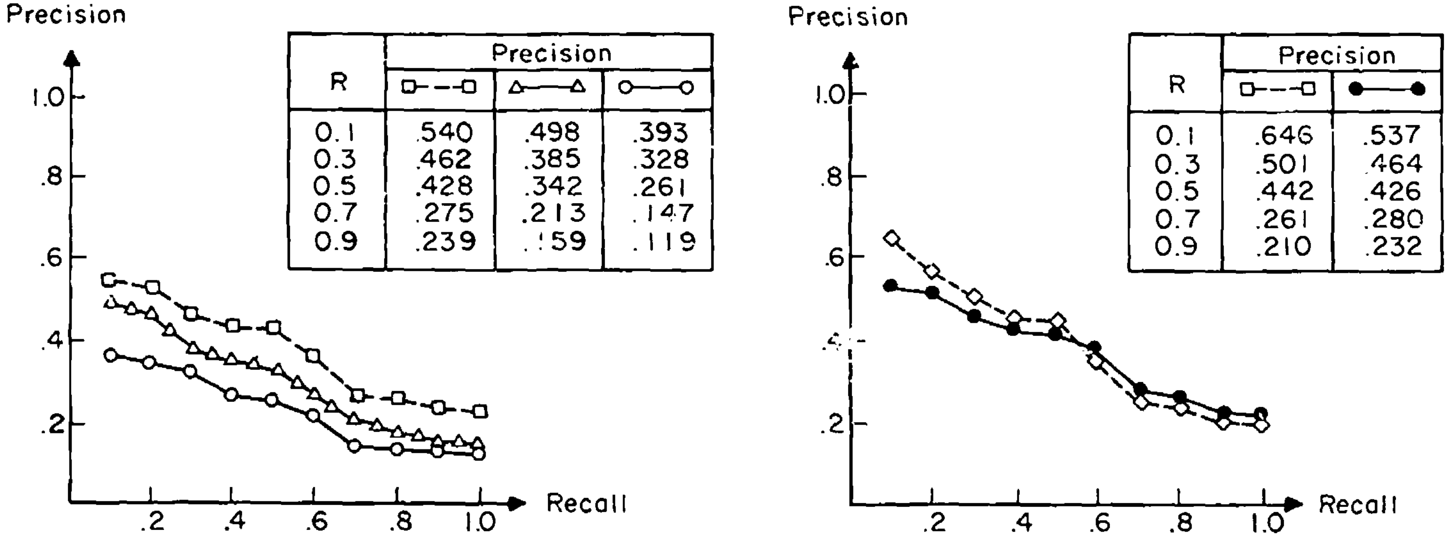
Fig. 4 shows the evaluation results obtained by using this process with a collection of eighty-two documents in the field of documentation, together with thirty-five user queries. A total of 1218 distinct word stems were initially available for the identification of documents. It is seen from Fig. 4(a) that the evaluation results verify the model completely:

a) as high frequency, nondiscriminators are first deleted, the space spreads out, and the corresponding recall-precision output (following deletion of 252 terms) is improved by about twenty percent;

b) when additional terms are deleted, the compactness of the space begins to increase as discriminators are removed, and the recall-precision performance deteriorates; the middle curve of Fig. 4(a) represents the performance following deletion of 1120 terms (in decreasing Q order), at which time the retrieval effectiveness has already diminished by about ten percent.

○—○ Full Stem Vectors
□--□ Full Minus 252 Terms
△—△ Full Minus 1120 Terms

◇--◇ Thesaurus
●—● Full Stem Minus 28 Terms



R	Precision □--□	Precision △—△	Precision ○—○
0.1	.540	.498	.393
0.3	.462	.385	.328
0.5	.428	.342	.261
0.7	.275	.213	.147
0.9	.239	.159	.119

a) Comparison of Various Deletion Levels

R	Precision □--□	Precision ●—●
0.1	.646	.537
0.3	.501	.464
0.5	.442	.426
0.7	.261	.280
0.9	.210	.232

b) Comparison with Thesaurus

Automatic Common Word Identification
(ADI Collection)

Fig. 4

A comparison between the standard thesaurus performance and a word stem method with the top twenty-eight common terms deleted is shown in Fig. 4(b). Is is seen that the thesaurus process is somewhat superior only at the low recall end with the two graphs being nearly equivalent over most of the performance region.

The results of Fig. 4 thus confirm the earlier studies of Fig. 2 in the sense that word stem matching methods produce performance parameters nearly equivalent to those obtainable by standard thesauruses, providing only that common word stems are appropriately identified, and removed as potential content identifiers.

3. Automatic Concept Grouping Procedures

For many years, the general classification problem consisting of the generation of groups, or classes, of items which are similar, in some sense, to each other has been of major concern in many fields of scientific endeavor. In information retrieval, documents are often classified by grouping them into clusters of items thereby simplifying the information search process. Alternatively, terms or concepts, are grouped into thesaurus classes in such a way that synonyms and other related terms are all identifiable by the same thesaurus class numbers.

In section 1 of this report, various criteria were specified for the manual, or intellectual construction of thesaurus classes. Since the manual generation of thesauruses requires, however, a great deal of time and experience, experiments have been conducted for some years leading to an automatic determination of thesaurus classes based on the properties of the available document collections, that is, on the assignment of terms to documents. The general process may be described as follows [11]:

a) a term-document matrix is first constructed specifying the assignment of terms to documents, including term weights, if any;

b) a term-term similarity matrix is generated from the term-document matrix by computing the similarity between each pair of term vectors, based on joint assignment of terms to documents;

c) a threshold value is applied to the term-term similarity matrix to produce a binary term-term connection matrix in which two terms are assumed connected (that is, a 1 appears in the connection matrix) whenever the similarity between corresponding term vectors is sufficiently high;

d) the binary connection matrix may be viewed as an abstract graph in which each term is represented by a node, and each existing connection as a branch between corresponding pairs of nodes; some function of this graph (for example, the connected components, or maximal complete sub-graphs of the graph) is then used to define the clusters, or classes of terms.*

A number of investigators have constructed term classifications automatically, using procedures similar to the ones outlined above [12, 13, 14]. Unfortunately, the generation of the term-term connection matrix is time-consuming and expensive when the number of terms is not very small. For this reason, less expensive automatic classification methods, in which

---

*A connected component of a graph is a subgraph for which each pair of nodes is connected by a path (a chain of branches); in a maximal complete subgraph, each pair of nodes is connected by a direct branch, and no node not in the subgraph will exhibit such a connection to all other nodes of the subgraph.

an existing rough classification is improved by selective modification of the original classes, tend to be used in practice. [15, 16]

To determine the effectiveness of such automatically constructed term classifications in a retrieval environment, three types of experiments are briefly described involving, respectively, an automatic refinement of already existing classes; two fully automatic term classification methods; and a semi-automatic classification process.

The first of these experiments consists in taking an existing term classification, or an existing thesaurus, and in refining the term classes by removing classes which are highly overlapping. [17] One such algorithm tried with the SMART system was based on the following steps (in addition to steps a) through d) already listed):

e) given the existing term classes, a class-class similarity matrix is constructed, using the procedures already outlined for the term-term matrix;

f) a threshold value is applied to the class-class matrix to produce a binary class connection matrix;

g) each maximal complete subgraph defines a new merged concept class;

h) merged classes that are subsets of other larger classes are removed, the remainder constituting the new merged classification.

This procedure was used to refine the documentation thesaurus originally available for the ADI collection, consisting of eighy-two documents and thirty-five search requests. Two "merged" thesauruses were produced as follows:

a) thesaurus 1 with a total of 156 concept classes and approximately 3.9 concepts per class;

b) thesaurus 2 with a total of 289 concept classes, averaging 1.4 concepts per class. [18]

The global normalized recall and precision values, averaged over the thirty-five queries and exhibited in Table 2, show that some improvement in performance is obtainable with the refining process.

The second, more ambitious group of experiments deals with the fully automatic classification procedures outlined at the beginning of this section. In one such study a large variety of graph theoretical definitions was used to define the term classes, including "strings of terms", "stars", "cliques", and "clumps", and various threshold and frequency restrictions were applied to the class generation methods. [19] In general, it is found that some of the automatic classifications operate more effectively than unclassified keywords, particularly if "strong" similarity connections (with a large threshold value) are used, and only nonfrequent terms are permitted to be classified. A comparison of the automatic classifications with manual thesauruses was not attempted in this case.

Another fully automatic term classification experiment was recently concluded, using procedures very similar to the preceding ones, with a large experimental collection of 11,500 document abstracts in computer engineering. [20] A class refining process was implemented in that case, and many different parameter variations were tried. In the end, only modest improvements were obtained over a standard word stem matching process, the author claiming that

Thesaurus Type	Normalized Recall	Normalized Precision
... Original Thesaurus	.800	.610
Merged Thesaurus 1	.830	.640
Merged Thesaurus 2	.830	.650

Merged Thesaurus Performance

Table 2

"in relation to results yielded by our various (automatic) associative strategies, it must be concluded that retrieval by the simple means of comparing keyword stems provides a very good level of performance." [20, p. 61]

The last term classification experiment is based on a semi-automatic method for generating the original term vectors used to produce the term-term similarity matrix. Specifically, a set of properties is manually generated by asking questions about each term, and properly encoding the answers.* For each term, the corresponding property vector is then defined as the set of answers obtained in response to ten or twelve manually generated questions. When all term vectors are available, one of the automatic classification procedures may be used to obtain the actual thesaurus classification. [3, 21]

Such a semi-automatic dictionary was constructed for documents in computer engineering. Its properties are compared with those of a manually constructed thesaurus in the summary of Table 3. It is seen that the semi-automatic thesaurus classes are much less homogeneous — some classes being very large, and some very small — than the corresponding manual classes. Furthermore, fewer common words are identified in the semi-automatic thesaurus.

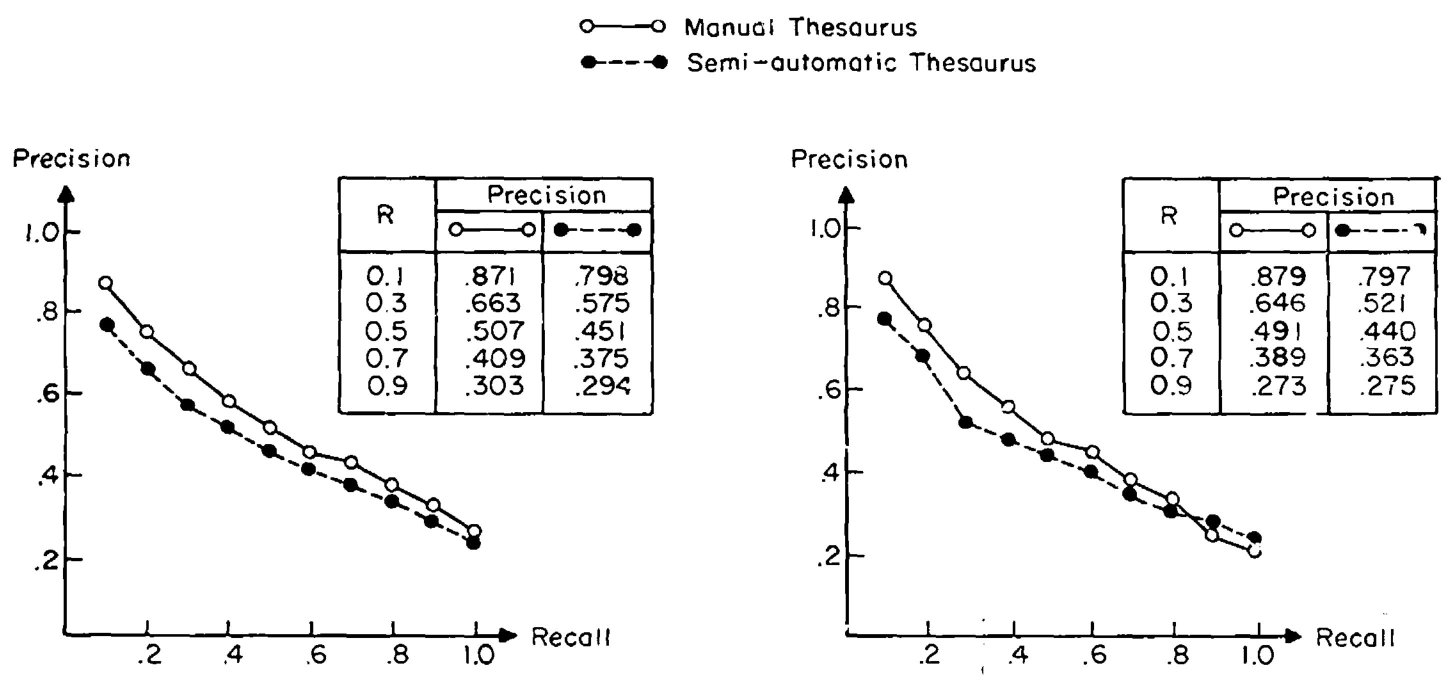
The retrieval results obtained with the two thesauruses are included in Fig. 5. It is seen that the semi-automatic thesaurus produces a less effective performance than the corresponding manually constructed dictionary

*A typical question might inquire whether a given term in computer science refers to computer hardware (1), or to computer software (2), or whether the question is inapplicable to the given term (3); the chosen answer is then encoded by the response number (n).

Properties	Manual (Harris) Thesaurus	Semi-Automatic (Bench) Thesaurus
Number of Concept Classes	863	2953
Number of Word (stem) Entries	2551	5197
Avg. Number of Words per Class	3	1.8
Number of Very Small (Single Word) Classes	468	2725
Number of Very Large Classes (32 to 101 Words)	2	12
Number of Words Appearing in Two or More Classes	52	275
Proportion of "Common" Words Compared to Total Words	37.3%	4.4%

Semi-Automatic Dictionary Properties

Table 3



a) ADI Text

R	Precision (Manual)	Precision (Semi-automatic)
0.1	.871	.798
0.3	.663	.575
0.5	.507	.451
0.7	.409	.375
0.9	.303	.294

b) ADI Abstract

R	Precision (Manual)	Precision (Semi-automatic)
0.1	.879	.797
0.3	.646	.521
0.5	.491	.440
0.7	.389	.363
0.9	.273	.275

Comparison of Manual and Semi-Automatic Thesaurus

Fig. 5

over most of the performance range. Only for very high recall is the effectiveness of both dictionaries approximately equal.

4. Summary

A number of manual and automatic dictionary construction procedures are described and evaluated in the present study, including in particular, automatic methods for the recognition of common words, and automatic or semi-automatic term grouping methods. It appears that the automatic common word recognition methodology can usefully be incorporated into existing text analysis systems; indeed, the effectiveness of the resulting extended word stem matching process appears equivalent to that obtainable with standard thesauruses.

The effectiveness of the automatic term grouping algorithms is still somewhat in doubt. The automatic grouping methods can probably be implemented more efficiently than the more costly manual thesaurus construction processes. However, no clearly superior automatic thesaurus, using term classes, has as yet been generated. [22, 23]

For the present time, a combination of manual and automatic thesaurus methods therefore appears most promising for practical applications, involving the following steps:

a) automatic common word recognition;

b) manual term classification;

c) automatic refining of the manually produced classes.

References

[1] G. Salton, Automatic Information Organization and Retrieval, McGraw-Hill Book Company, New York, 1968.

[2] E. Wall, Vocabulary Building and Control Techniques, American Documentation, Vol. 20, No. 2, April 1969, p. 161-164.

[3] G. Salton and M. E. Lesk, Information Analysis and Dictionary Construction, Scientific Report No. ISR-11 to the National Science Foundation, Section IV, Dept. of Computer Science, Cornell University, June 1966.

[4] M. E. Lesk, Performance of Automatic Information Systems, Information Storage and Retrieval, Vol. 4, 1968, p. 201-218.

[5] G. Salton, Computer Evaluation of Indexing and Text Processing, Journal of the ACM, Vol. 15, No. 1, January 1968, p. 8-36.

[6] A. Moser, Construction of Dictionaries for Text Analysis and Retrieval, unpublished manuscript, Cornell University, 1970.

[7] M. Coyaud and N. Siot-Decauville, L'Analyse Automatique des Documents, Mouton and Co., Paris 1967.

[8] E. B. Fedorov and V. S. Cherniavskii, Automatic Translation from a Natural Language into the Descriptive Language of Systems of the "Empty-Nonempty" Type, in Systems for Handling Information, Moscow, 1969.

[9] D. Bergmark, The Effect of Common Words on Retrieval Performance, Scientific Report No. ISR-18 to the National Science Foundation and to the National Library of Medicine, Dept. of Computer Science, Cornell University, October 1970.

[10] K. Bonwit and J. Aste-Tonsman, Negative Dictionaries, Scientific Report No. ISR-18 to the National Science Foundation and to the National Library of Medicine, Dept. of Computer Science, Cornell University, October 1970.

[11] J. G. Augustson and J. Minker, An Analysis of Some Graph Theoretical Cluster Techniques, Journal of the ACM, Vol. 17, No. 4, October 1970, p. 571-588.

[12] H. Borko, The Construction of an Empirically Based Mathematically Derived Classification System, Report No. SP-588, System Development Corporation, Santa Monica, October 1961.

[13] S. F. Dennis, The Design and Testing of a Fully Automatic Index Searching System for Documents Consisting of Expository Text, in G. Schecter, editor, "Information Retrieval — A Critical View", Thompson Book Co., Washington, D. C., 1967.

[14] K. Sparck Jones and D. Jackson, Current Approaches to Classification and Clump Finding, Computer Journal, Vol. 10, No. 1, May 1967, p. 29-37.

[15] L. B. Doyle, Breaking the Cost Barrier in Automatic Classification Report No. SP-2516, System Development Corp., Santa Monica, July 19[illegible].

[16] R. T. Dattola, A Fast Algorithm for Automatic Classification, Scientific Report No. ISR-16 to the National Science Foundation, Section V, Computer Science Dept., Cornell University, October 1968.

[17] C. C. Gotlieb and S. Kumar, Semantic Clustering of Index Terms, Journal of the ACM, Vol. 15, No. 4, October 1968, p. 493-513.

[18] R. T. Dattola and D. M. Murray, An Experiment in Automatic Thesaurus Construction, Scientific Report No. ISR-13 to the National Science Foundation, Section VIII, Dept. of Computer Science, Cornell University, December 1967.

[19] K. Sparck Jones and D. M. Jackson, The Use of Automatically-Obtained Keyword Classifications for Information Retrieval, Information Storage and Retrieval, Vol. 5, No. 4, February 1970, p. 175-202.

[20] P. K. T. Vaswani and J. B. Cameron, The NPL Experiments in Statistical Word Associations and their Use in Document Indexing and Retrieval, Report Com. Sci. 42, National Physical Laboratory, Teddington, England, April 1970.

[21] G. Salton, Information Dissemination and Automatic Information Systems, Proc. IEEE, Vol. 54, No. 12, December 1966, p. 1663-1678.

[22] C. W. Cleverdon and E. M. Keen, Factors Determining the Performance of Indexing Systems, Aslib-Cranfield Research Project Report, Vol. 1 and 2, Cranfield, England, 1966.

[23] G. Salton, Automatic Text Analysis, Science, Vol. 168, 17 April 1970, p. 335-343.

1. Compactness Function

$$\boxed{Q = \sum_{j=1}^{N} \cos(c, v_j)} \qquad 0 \le Q \le N$$

where

N = total number of documents

c = centroid of document space

v_j = concept vector for document j

2. Term Deletion Algorithm

Let $Q_i = Q$ with term i deleted

If $Q_i > Q$ → document space is more compact
term i is a discriminator

If $Q_i < Q$ → document space is more spread out
term i is a nondiscriminator

Delete set of terms I such that Q_I is minimal

Automatic Common Word Recognition