DOCUMENT RESUME

ED 048 911 LI 002 720

TITLE	Automatic Content Analysis; Part 1 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 I]
PUB DATE	Oct 70
NOTE	169p.; Part of LI 002 719
EDRS PRICE	EDRS Price MF-$0.65 HC-$6.58
DESCRIPTORS	*Automatic Indexing, Automation, Bibliographic Citations, *Content Analysis, Electronic Data Processing, *Evaluation, Indexing, *Information Retrieval, Lexicology, Programing Languages, *Relevance (Information Retrieval), Vocabulary
IDENTIFIERS	Automatic Content Analysis, On Line Retrieval Systems, *Saltons Magical Automatic Retriever of Texts, SMART

ABSTRACT

Four papers are included in Part One of the eighteenth report on Salton's Magical Automatic Retriever of Texts (SMART) project. The first paper: "Content Analysis in Information Retrieval" by S. F. Weiss presents the results of experiments aimed at determining the conditions under which content analysis improves retrieval results as well as the degree of improvement obtained. The second paper: "The 'Generality' Effect and the Retrieval Evaluation for Larger Collections" by G. Salton assesses the role of the generality effect in retrieval system evaluation and gives evaluation results for the comparisons of several document collections of distinct size and generality in the areas of documentation and aerodynamics. In the third paper: "Automatic Indexing Using Bibliographic Citations" by G. Salton citations are used directly to identify document content and an attempt is made to evaluate their effectiveness in a retrieval environment. The final paper: "Automatic Resolution of Ambiguities from Natural Language Text" by S. F. Weiss discusses the evolutionary process by which ambiguities are created and classifies ambiguities into three classes: true, contextual and syntactic. (For the entire SMART project report see LI 002 719, for parts 2-5 see LI 002 721 through LI 002 724.) (NH)

Department of Computer Science

Cornell University

Ithaca, New York 14850

Automatic Content Analysis

Part I

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

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 One are reproduced here as LI 002 720. See LI 002 721 thru LI 002 724 for Parts 2 - 5.

TABLE OF CONTENTS

Page

SUMMARY xv

PART ONE

AUTOMATIC CONTENT ANALYSIS

I. WEISS, S. F.
"Content Analysis in Information Retrieval"

Abstract I-1

1. Introduction I-2

2. ADI Experiments I-5

A) Statistical Phrases I-5

B) Syntactic Phrases I-7

C) Cooccurrence. I-9

D) Elimination of Phrase List I-12

E) Analysis of ADI Results I-20

3. The Cranfield Collection I-26

4. The TIME Subset Collection. I-27

A) Construction. I-27

B) Analysis of Results I-31

5. A Third Collection I-39

6. Conclusion I-43

References I-46

II. SALTON, G.
"The 'Generality' Effect and the Retrieval Evaluation for Large Collections"

TABLE OF CONTENTS (continued)

Page

II. continued

Abstract II-1

1. Introduction. II-1

2. Basic System Parameters II-3

3. Variations in Collection Size II-7

A) Theoretical Considerations II-7

B) Evaluation Results II-10

C) Feedback Performance II-15

4. Variations in Relevance Judgments. II-24

5. Summary II-31

References. II-33

III. SALTON, G.
"Automatic Indexing Using Bibliographic Citations"

Abstract III-1

1. Significance of Bibliographic Citations. III-1

2. The Citation Test III-4

3. Evaluation Results. III-9

References. III-19

Appendix III-20

IV. WEISS, S. F.
"Automatic Resolution of Ambiguities from Natural Language Text"

TABLE OF CONTENTS (continued)

Page

IV. continued

Abstract IV-1

1. Introduction. IV-2

2. The Nature of Ambiguities IV-4

3. Approaches to Disambiguation IV-8

4. Automatic Disambiguation. IV-14

A) Application of Extended Template Analysis to Disambiguation IV-14

B) The Disambiguation Process IV-15

C) Experiments IV-17

D) Further Disambiguation Processes IV-20

5. Learning to Disambiguate Automatically IV-21

A) Introduction IV-21

B) Dictionary and Corpus IV-21

C) The Learning Process IV-23

D) Spurious Rules IV-28

E) Experiments and Results IV-30

F) Extensions. IV-46

6. Conclusion IV-49

References. IV-50

PART TWO

AUTOMATIC DICTIONARY CONSTRUCTION

Available as LI 002 721

V. BERGMARK, D.

TABLE OF CONTENTS (continued)

Page

V. continued

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

Abstract V-1

1. Introduction. V-1

2. Experiment Outline. V-2

   A) The Experimental Data Base V-2
   B) Creation of the Significant Stem Dictionary. . . . V-2
   C) Generation of New Query and Document Vectors . . . V-4
   D) Document Analysis — Search and Average Runs. . . . V-5

3. Retrieval Performance Results V-7

   A) Significant vs. Standard Stem Dictionary. V-7
   B) Significant Stem vs. Thesaurus V-9
   C) Standard Stem vs. Thesaurus V-11
   D) Recall Results V-11
   E) Effect of "Query Wordiness" on Search Performance. . V-15
   F) Effect of Query Length on Search Performance . . . V-15
   G) Effect of Query Generality on Search Performance . . V-17
   H) Conclusions of the Global Analysis. V-19

4. Analysis of Search Performance. V-20

5. Conclusions V-31

6. Further Studies. V-32

References. V-34

Appendix I. V-35

Appendix II V-39

TABLE OF CONTENTS (continued)

Page

VI. BONWIT, K. and ASTE-TONSMANN, J.
"Negative Dictionaries"

Abstract VI-1

1. Introduction. VI-1

2. Theory. VI-2

3. Experimental Results VI-7

4. Experimental Method VI-19

A) Calculating ρ_i VI-19

B) Deleting and Searching. VI-20

5. Cost Analysis VI-25

6. Conclusions VI-29

References. VI-33

VII. SALTON, G.
"Experiments in Automatic Thesaurus Construction for Information Retrieval"

Abstract VII-1

1. Manual Dictionary Construction. VII-1

2. Common Word Recognition VII-8

3. Automatic Concept Grouping Procedures VII-17

4. Summary VII-25

References. VII-26

TABLE OF CONTENTS (continued)

Page

PART THREE

USER FEEDBACK PROCEDURES

Available as LI 002 722

VIII. BAKER, T. P.
"Variations on the Query Splitting Technique with Relevance Feedback"

Abstract VIII-1

1. Introduction. VIII-1

2. Algorithms for Query Splitting. VIII-3

3. Results of Experimental Runs VIII-11

4. Evaluation VIII-23

References. VIII-25

IX. CAPPS, B. and YIN, M.
"Effectiveness of Feedback Strategies on Collections of Differing Generality"

Abstract IX-1

1. Introduction. IX-1

2. Experimental Environment. IX-3

3. Experimental Results IX-8

4. Conclusion IX-19

References. IX-23

Appendix IX-24

TABLE OF CONTENTS (continued)

Page

X. KERCHNER, M.
"Selective Negative Feedback Methods"

Abstract X-1

1. Introduction. X-1

2. Methodology X-2

3. Selective Negative Relevance Feedback Strategies. . . X-5

4. The Experimental Environment X-6

5. Experimental Results X-8

6. Evaluation of Experimental Results X-13

References. X-20

XI. PAAVOLA, L.
"The Use of Past Relevance Decisions in Relevance Feedback"

Abstract XI-1

1. Introduction. XI-1

2. Assumptions and Hypotheses XI-2

3. Experimental Method XI-3

4. Evaluation XI-7

5. Conclusion XI-12

References. XI-14

10

Page

PART FOUR

CLUSTERING METHODS

Available as LI 002 723

XII. JOHNSON, D. B. and LAFUENTE, J. M.
"A Controlled Single Pass Classification Algorithm with Application to Multilevel Clustering"

Abstract XII-1

1. Introduction. XII-1

2. Methods of Clustering. XII-3

3. Strategy XII-5

4. The Algorithm XII-6

   A) Cluster Size XII-8

   B) Number of Clusters XII-9

   C) Overlap. XII-10

   D) An Example. XII-10

5. Implementation XII-13

   A) Storage Management XII-14

6. Results XII-14

   A) Clustering Costs. XII-15

   B) Effect of Document Ordering XII-19

   C) Search Results on Clustered ADI Collection . . . XII-20

   D) Search Results of Clustered Cranfield Collection . XII-31

7. Conslusions XII-34

References. XII-37

TABLE OF CONTENTS (continued)

Page

XIII. WORONA, S.
"A Systematic Study of Query-Clustering Techniques: A Progress Report"

Abstract XIII-1

1. Introduction. XIII-1

2. The Experiment XIII-4

A) Splitting the Collection XIII-4

B) Phase 1: Clustering the Queries XIII-6

C) Phase 2: Clustering the Documents. XIII-8

D) Phase 3: Assigning Centroids XIII-12

E) Summary. XIII-13

3. Results XIII-13

4. Principles of Evaluation. XIII-16

References. XIII-22

Appendix A. XIII-24

Appendix B. XIII-29

Appendix C. XIII-36

PART FIVE

ON-LINE RETRIEVAL SYSTEM DESIGN

Available as LI 002 724

XIV. WILLIAMSON, D. and WILLIAMSON, R.
"A Prototype On-Line Document Retrieval System"

Abstract XIV-1

12

TABLE OF CONTENTS (continued)

Page

XIV. continued

1. Introduction. XIV-1

2. Anticipated Computer Configuration XIV-2

3. On-Line Document Retrieval — A User's View. XIV-4

4. Console Driven Document Retrieval — An Internal View . XIV-10

   A) The Internal Structure. XIV-10
   B) General Characteristics of SMART Routines . . . XIV-16
   C) Pseudo-Batching XIV-17
   D) Attaching Consoles to SMART XIV-19
   E) Console Handling — The Superivsor Interface . . XIV-21
   F) Parameter Vectors XIV-21
   G) The Flow of Control. XIV-22
   H) Timing Considerations XIV-23
   I) Noncore Residenc Files. XIV-26
   J) Core Resident Files. XIV-28

5. Consol — A Detailed Look. XIV-30

   A) Competition for Core XIV-30
   B) The SMART On-line Console Control Block XIV-31
   C) The READY Flag and the TRT Instruction XIV-32
   D) The Routines LATCH, CONSIN, and CONSOT XIV-32
   E) CONSOL as a Traffic Controller XIV-34
   F) A Detailed View of CYCLE XIV-37

6. Summary XIV-39

Appendix XIV-40

XV. WEISS, S. F.
"Template Analysis in a Conversational System"

TABLE OF CONTENTS (continued)

Page

XV. continued

Abstract XV-1

1. Motivation XV-1

2. Some Existing Conversational Systems. XV-4

3. Goals for a Proposed Conversational System. . . XV-7

4. Implementation of the Conventional System . . . XV-11

A) Capabilities XV-11

B) Input Conventions XV-12

C) The Structure of the Process. XV-13

D) Template Analysis in the Conversational System XV-14

E) The Guide Facility XV-23

F) Tutorials XV-24

5. Experimentation. XV-25

A) System Performance XV-30

B) User Performance. XV-31

C) Timing XV-34

6. Future Extensions XV-35

7. Conclusion XV-37

References. XV-39

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 One are reproduced here as LI 002 720. See LI 002 721 thru LI 002 724 for Parts 2 - 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 environ-

ment 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

I. Content Analysis in Information Retrieval

S. F. Weiss

Abstract

In information retrieval there exist a number of content analysis schemes which analyze natural language text to varying degrees of complexity. Regardless of how well the text analysis is performed by each process, the true value of a given process lies in its effectiveness as an information retrieval tool. The performance may in each case be investigated by actual retrieval tests using the various proposed content analysis schemes.

Results obtained with a variety of linguistic phrase recognition methods show that very little, if any, improvements in retrieval effectiveness are obtained when any of the refined content analysis schemes are used with existing document collections. The main reason appears to be the fact that the value of refined content analysis systems resides in their effectiveness in separating lexically similar, but semantically different documents. Existing collections are too sparse, and do not contain many close documents. When denser collections are created, it can be shown that linguistic content analysis methods become of increasing value as the density increases. The queries also influence the type of content analysis to be used. In general, queries of the question-answering variety show improved retrieval results with increasing refinements in the content analysis. Document retrieval queries do not exhibit this type of improvement.

Future work must be devoted to a determination of what makes a user judge a particular document to be relevant. With more insight into the relevance area, the role of linguistic content analysis in information retrieval may become more clearly defined.

1. Introduction

The purpose of a content analysis system as considered in this study is as an information retrieval aid. It is therefore necessary to perform retrieval using various content analysis methods to determine how well it fulfills its actual role. This study presents experiments and results aimed at determining the conditions under which content analysis improves retrieval results as well as the degree of improvement obtained. All information retrieval systems use some degree of content analysis in its broadest sense. This is generally in the form of assignment of concept indicators to individual words. But in this study content analysis refers to the analysis and utilization of multi-word groups as information retrieval tools.

Using phrases determined by content analysis as an information retrieval aid is theoretically very appealing. It adds another dimension to search capabilities beyond the single word matching used by most infromation retrieval systems. Documents and queries are matched not only on content, but on the interrelationship of content elements as well. Hutchins [3] has proposed an information retrieval system based solely on the cooccurrence of phrases in documents and queries. However, some experiments indicate that phrases alone may be too strict a criterion for useful results. A more reasonable approach is to use phrases in conjunction with a less structured method such as word or concept matching. Therefore in this study phrases are considered as an adjunct to single concept matching.

A number of existing information retrieval systems permit searching on multi-word structured information. Some systems such as that ...gned by Curtice and Jones at Arthur D. Little [1] index documents

and queries by contiguous word pairs as well as individual words. Retrieval is thus aided by this rudimentary form of phrase analysis. The IBM Document Processing System [4] takes this capability one step further. Multi-word search keys can be specified using a number of options besides simple contiguity. For example, consider the sample queries below. Query A retrieves documents containing "information" and "retrieval" in that order and separated by at most one other word. Query B retrieves documents with the same two words separated by at most one word but with no restriction on ordering. This will retrieve "information retrieval" as well as "retrieval of information". Queries C and D further relax the proximity criterion and retrieve documents in which "information" and "retrieval" occur within the same sentence and the same paragraph respectively.

A. INFORMATION RETRIEVAL (+1)

B. INFORMATION RETRIEVAL (-+1)

C. INFORMATION RETRIEVAL (SEN)

D. INFORMATION RETRIEVAL (PAR)

This specification is an attempt to perform some degree of semantic normalization. It permits the association of phrases which are semantically similar but structurally different. However the IBM system and others like it approach the semantic normalization by structural rather than semantic means. The resultant semantic processes are hence necessarily very superficial. As Lesk points out, phrases determined by processes of this type may cooccur in documents and queries too infrequently for them to be of any practical value. Lesk therefore proposes an information retrieval system in which documents and queries are subjected to a complex syntactic semantic analysis. Phrase normalization is then based on meaning rather

than just structure [5]. A few other semantically based content analysis schemes exist such as the manual indexing process developed by Mandersloot, Douglas and Spicer [2]. Of all existing information retrieval systems with content analysis capabilities, the SMART system provides the greatest variety of content analysis methods. This makes SMART an excellent experimental facility for testing content analysis in general. The various SMART content analysis methods are presented in some detail later in this study.

In information retrieval, phrases can do two things. First, they can distinguish between two documents with similar content elements but different meaning. For example, the two inputs below are assigned identical concept vectors by normal text cracking methods. To distinguish between them requires that the structure as well as the content of the input be considered.

A. Design of computer systems

B. Computerized design systems

A second job performed by phrases is that of reinforcing correlations between queries and documents which have similar phrases. In this way the cooccurrence in the document and query of concepts which form a phrase is weighted more heavily than the cooccurrence of a similar number of unrelated concepts. While this might appear to be a convincing case in favor of using phrases in information retrieval, the previous argument is purely theoretical. It remains to test the theory by performing retrieval using various phrase determination methods. It is necessary to analyze the results obtained not only to determine how the overall results compare with those achieved without the use of phrases, but also to determine the exact cause

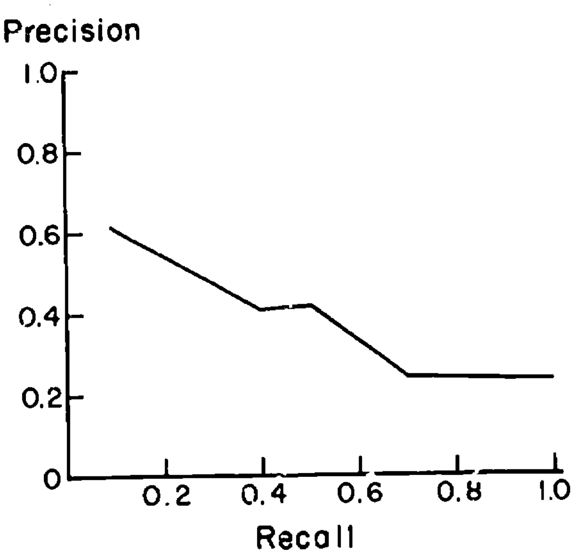
of the phrase method results. That is, are the new results a function of the document or query collections used, the phrase determining technique, the matching procedure, or a combination of several factors?

2. ADI Experiments

The first set of experiments uses the ADI collection. This is a set of eighty-two documents and thirty-five queries in the field of documentation. About half of the queries ask for specific information while the other half are of a more general nature. A set of ten queries, five general and five specific, is chosen as representative of the various query forms and constructions. A normal SMART retrieval run is then performed on the entire ADI collection and the ten test queries. For each query the ten most highly correlated documents are identified. These documents along with any others, relevant to the test queries but not in the top ten, are collected to form a test document set. The total set contains 56 of the 82 ADI documents. In all the experiments phrases are determined for this test set only. It is felt that the results achieved with this limited set will differ little from those of the full set. The use of a restricted set such as this is also a practical necessity since the great quantity of hand analysis required by these experiments precludes the use of the full document and query sets. Figure 1 indicates the results of a normal cosine retrieval process using the ten test queries. The following subsections discuss experimentation using various phrase determining techniques.

A) Statistical Phrases

The statistical phrase process uses a predetermined list of phrases.



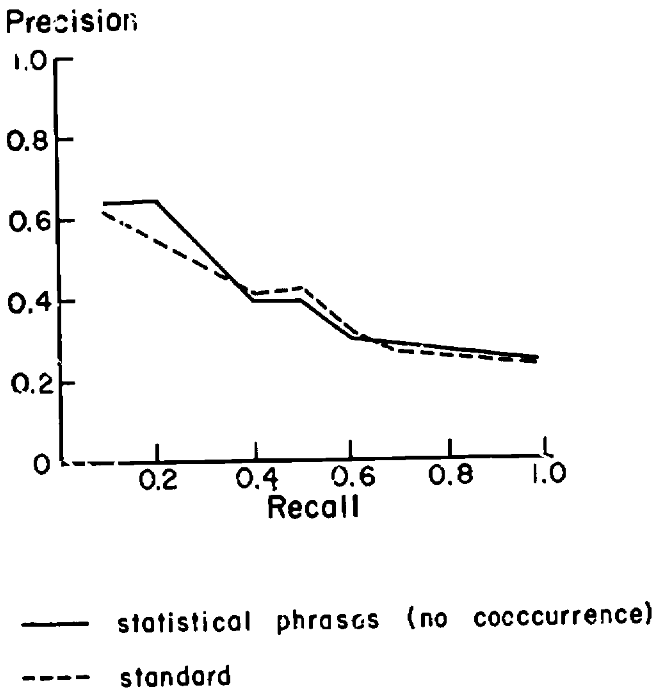
Standard Smart Results
(No Phrases)

Figure 1

The occurrence of the phrase elements in a document or query is considered an occurrence of that phrase regardless of the syntactic relation of the phrase components. A concept number is associated with a phrase and the appropriate concepts are appended to the document or query vectors. This method is clearly the simplest way to determine phrases since it requires no syntactic analysis of the text. However, statistical phrases have some serious drawbacks. Most obvious is the fact that they may recognize false phrases; that is, occurrences of the desired phrase elements but not in the proper syntactic relation. This problem can be minimized in small collections dealing with a narrow subject area by judicious selection of the statistical phrase list. In a corpus dealing with computer systems, for example, the occurrence of the words "real" and "time" can be viewed with relative certainty to be an occurrence of the phrase "real time". However as the collection grows and the subject area broadens, these decisions become less certain. Also the difficulty in creating the phrase list is increased as the corpus is enlarged. The phrase list can be determined by statistical means; however, weaknesses in this method can create problems. In the ADI collection for example, of the 409 statistical phrases in the test document set, only 153, roughly 37%, are syntactically correct. Figure 2 shows the results achieved using statistical phrases along with the standard no-phrase results. The results for statistical phrases are slightly higher in places, lower in others and show no significant overall improvement in retrieval quality.

B) Syntactic Phrases

As mentioned previously, almost two-thirds of the statistical phrases determined for the test set turn out to be syntactically incorrect. Removal



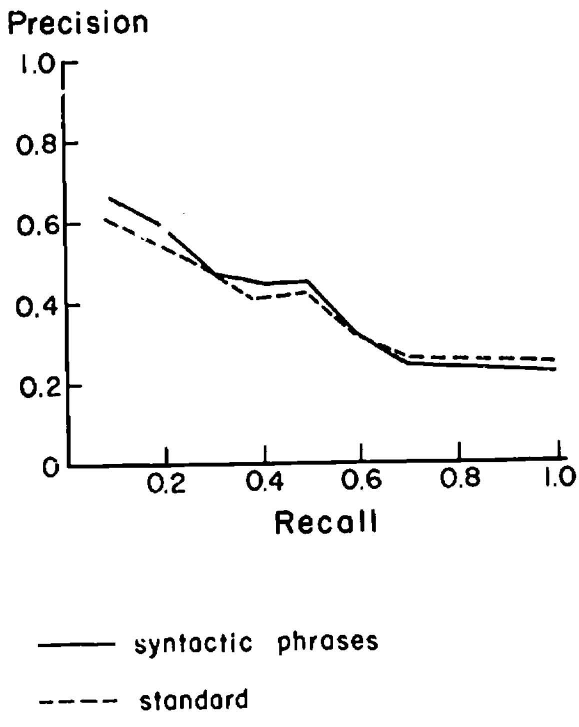
Experiment 2: Statistical Phrases

Figure 2

of the false phrases would allow the phrase component of the concept vector to represent more closely the true structure of the document or query. An automated process to perform this would first locate statistical phrases and then, using some syntactic analysis technique, weed out the erroneous ones. The syntactic analysis process required here is considerably simpler than general syntactic analysis since the process need only check the correctness of a statistical phrase rather than perform a complete syntactic parse. However, since the purpose of this study is to determine the value of syntactic phrases as a retrieval aid and not to test a syntactic analyzer, the analyses are done by hand. Removal of false phrases leaves 153 of the original 409 document phrases and 6 of the 12 query phrases. Results of this process are presented in Figure 3, and are again, disappointing. Statistical phrases show no significant improvement in retrieval performance.

C) Cooccurrence

The easiest way to handle phrases, and the way used in the previous experiments, is simply to assign each phrase a concept number and append the number onto the appropriate concept vector. After assignment, phrase concepts become indistinguishable from single word concepts, and the correlation coefficient operates normally. Unfortunately this gives rise to a number of serious problems. First, is the dilution effect caused by unmatched phrase concepts. The probability of a phrase match between a document and query is quite small due to the added structural requirements inherent in phrase matching. Furthermore since documents are typically much longer than queries, the document contains many phrases which cannot possibly match the query. As a consequence many phrase concepts are not matched. These unmatched concepts lower the correlation and partially if



Experiment 3: Syntactic Phrases
(No Cooccurrence)
Figure 3

not completely offset any gain achieved by matched phrases. Thus the inclusion of too many phrases can dilute the vector with unusable information and inferior results may be produced.

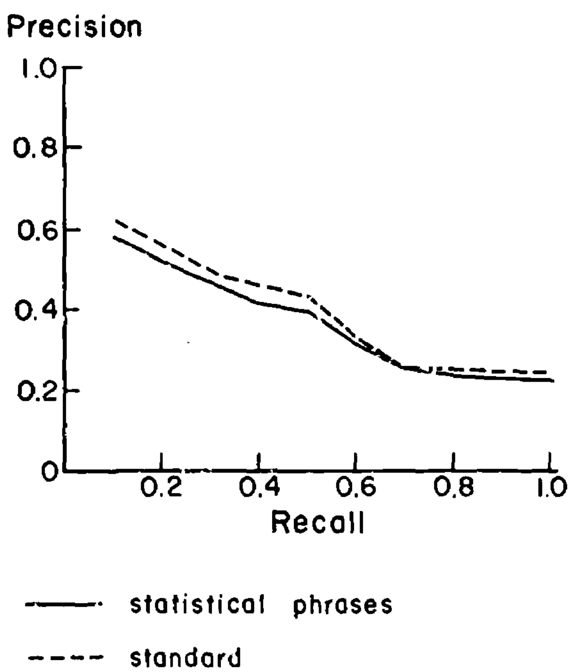
A second problem deals with the value of a phrase as a nonrelevancy indicator. Individual word concepts are about equal as relevancy and nonrelevancy indicators. That is the cooccurrence of concept A in document D and query Q is as good a measure of D's relevance to Q as the lack of this cooccurrence is a measure of D's nonrelevance. As more structure is imposed on the comparison of documents and queries, cooccurrences become more significant but less frequent while non-cooccurring structures become less significant and more frequent. For example if documents are retrieved only if they match, word for word, the complete query, few if any documents would be returned. However any document which is retrieved by this scheme would almost certainly be relevant. On the other hand, the fact that some documents do not match the complete query is not a good indicator of their nonrelevance. The situation is similar for phrases. Thus treating phrase concepts simply as additional word concepts over-emphasizes their role as nonrelevancy indicators and while it may provide improved precision, it has disastrous effects on recall.

The problems presented above make it necessary to treat phrase and word concepts differently. In particular the role of phrases as a relevancy indicator must be weighted much more heavily than their role as a nonrelevancy indicator. The method designed to accomplish this is called <u>cooccurrence matching</u> and considers phrases only when they cooccur between a document and a query. Its operation may be seen from the following example. Let D and Q be the word concept vectors for a particular document and query, and PD and PQ, their associated phrase concept vectors. If phrase concepts are

treated as word concepts, the correlation is calculated between D + PD and Q + PQ. The cooccurrence method on the other hand first calculates C = PQ $\cap$ PD. That is, C is the set of phrase concepts common to both the query and document. Correlation is then calculated between D + C and Q + C. In this way it is guaranteed that phrase concepts cannot lower the correlation, and in the worst case where C is empty, the correlation is unaffected by the phrases. This process avoids the two previously discussed pitfalls associated with phrase use. First, by ignoring all unmatched phrase concepts, the vectors cannot become diluted with useless and possibly detrimental information. Secondly, phrases are used only as a relevancy indicator while their far weaker role of nonrelevancy indicator is not considered. The experiments performed in the remainder of this study all employ the cooccurrence principle for handling phrase concepts. The next two experiments are repeats of the previous two with the addition of the use of the cooccurrence phrase matching technique. The results are shown in Figures 4 and 5 and once again show no improvement over the no phrase method. A more complete analysis of these results is presented below.

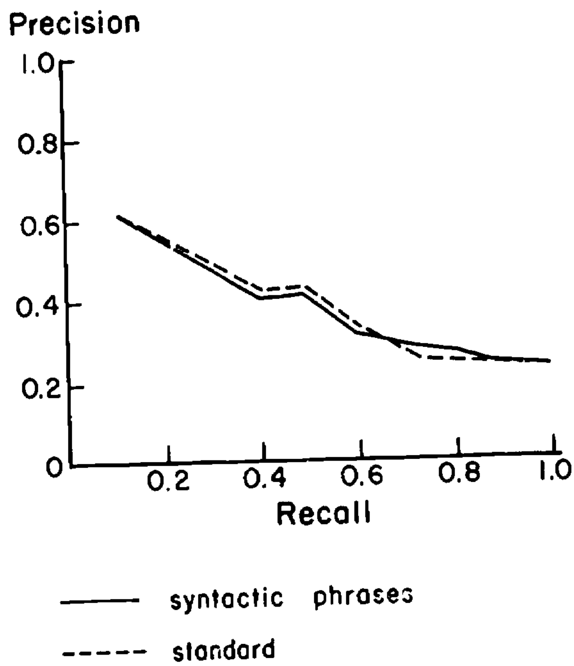
D) Elimination of the Phrase List

All methods discussed so far for using phrases in retrieval have required a phrase list. As previously mentioned the creation of these lists, whether by hand or by statistical processes, raises certain inherent problems. In general, it is far more desirable to be able to determine phrases without the need of such a list. One possible solution is to perform a syntactic analysis of the text, and determine all the phrases. The set of phrases thus generated is then normalized to associate all



Experiment 4: Statistical Phrases
(With Cooccurrence)

Figure 4



Experiment 5: Syntactic Phrases
(With Cooccurrence)
Figure 5

syntactically different but semantically identical phrases. This is accomplished, for example, by transformational kernelization of the phrases or by the use of a criterion tree matching scheme. Each phrase in the reduced set is then assigned a concept number, and retrieval proceeds as in the previous cases. However the syntactic analysis and normalization processes are prohibitively complex and produce a very large number of phrases. For these reasons an alternate method is used.

One of the easiest ways of accomplishing some degree of phrase processing without a phrase list is by means of the implicit phrase method. The philosophy behind this technique is that the cooccurrence in the document and query of several different concepts should be considered a better relevancy indicator than the cooccurrence of a single concept which has multiple occurrences and hence a higher weight. Consider the sample query and document vectors in Figure 6. The cosine correlation assigns the same correlation value to both. The second document however would seem to be more relevant to the query. The use of implicit phrases allows this fact to be reflected in the final correlation value. The basis of this process is a modified correlation coefficient formula:

$$C_{dq} = \frac{\sum_{i=1}^{N} d_i q_i + K(m-1)}{\left\{\left[\sum_{i=1}^{N} d_i^2 + K(m-1)\right]\left[\sum_{i=1}^{N} q_i^2 + K(m-1)\right]\right\}}$$

where m is the number of different concepts which cooccur in the document and query, and K is a constant. In the general case K = .4*P where P is an experimental parameter. In this way each pair of cooccurring concepts in the document and query is treated as a phrase and the correlation is is treated accordingly. In Figure 6 for example, the implicit phrases

QUERY: INFORMATION RETRIEVAL

DOC-1 INFORMATION ABOUT INFORMATION
DOC-2 INFORMATION RETRIEVAL AND SYSTEMS ANALYSIS

VECTORS:

	INFORMATION	RETRIEVAL	SYSTEMS	ANALYSIS	CORRELATION WITH QUERY
QUERY	12	12			
DOC-1	24				0.786
DOC-2	12	12	12	12	0.786

Sample Document and Query Vectors

Figure 6

correlation between document 1 and the query remains unchanged while the correlation of document 2 is raised to 0.774 thus reflecting its apparent greater relevancy. Figure 7 shows the results of retrieval using the ADI collection and the implicit phrase process with various values for P. It indicates that some improvement is achieved over the no-phrase process. However, one of the main drawbacks of the process is that it fails to fulfill one of the primary objectives for phrase use. That is it cannot discriminate between documents with similar concepts but different structural relationships among these concepts. For this reason a more syntactically oriented approach to phrase processing must be used.

The syntactic process used is relational content analysis. This process determines syntactic relations between pairs of text words. The details of relational content analysis are discussed by Weiss [9]. Concepts which are determined to be related by the content analyzer are encoded into a special phrase concept number, XXXXYYYYZZ, where XXXX represents the concept number of the first word, YYYY the second, and ZZ is the relation between them. The order of the two concepts is significant for all relations except parallel in which the smaller concept number appears first. The encoded relational phrases are treated as concept numbers and assembled into a phrase concept vector. The phrase vector must be kept separate from the word vector to permit the use of the cooccurrence phrase matching process. The retrieval results for this technique with the ADI test set appear in Figure 8.

Using this type of process for phrase determination has a number of advantages. First, it alleviates the need for an _a priori_ phrase list. Also, being a relatively simple process, it has significantly more practical value than some of the more complex systems. Clearly a great deal of

RECALL	IMPLICIT PHRASE TRIAL			
	1	2	3	4
0.1	.6124	.6333+	.6310+	.6278+
0.2	.5524	.6333+	.6310+	.6278+
0.3	.4862	.5643+	.5643+	.5577+
0.4	.4286	.4392+	.4356+	.4291+
0.5	.4335	.4309-	.4273-	.4255-
0.6	.3351	.3441+	.3341-	.3341-
0.7	.2608	.2651+	.2565-	.2538-
0.8	.2569	.2682+	.2597+	.2570+
0.9	.2493	.2590+	.2549+	.2427-
1.0	.2493	.2590+	.2549+	.2427-

1. = standard, no phrases

2. = implicit, p=1.0

3. = implicit, p=1.5

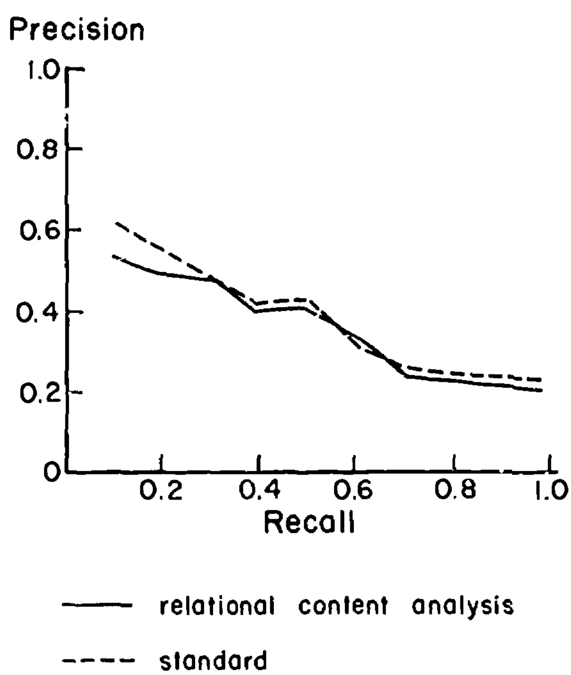
4. = implicit, p=2.0

+ indicates better than trial 1

- indicates worse than trial 1

ADI with Implicit Phrases

Figure 7



Experiment 7: Relational Content Analysis

Figure 8

syntactic information is lost since only word pairs are considered however, cooccurrences in documents and queries of syntactic structures more complex than word pairs is exceedingly rare. Thus despite its simplicity, relational content analysis does perform the particular aspect of syntactic analysis most relevant to information retrieval. Besides the advantages there are also some disadvantages inherent in this type of system. Most serious is its inability to associate semantically similar phrases. A system that uses a phrase list can recognize equivalent phrases whose constituent concepts are not equivalent. For example, the phrases "memory holding" and "data processing" are both assigned the same phrase concept by the SMART phrase list for the ADI collection, while each of the four words falls into a different concept class. The recognition of such equivalent phrases is impossible for systems which do not employ such a list of extensive semantic normalization. It may therefore be expected that retrieval results achieved by the relational concept analyzer will be inferior to those achieved in previous experiments. However, retrieval without the requirement of a phrase list seems to be a more reasonable approach to the problem. This is especially true in the case of large document collections where manual creation of a phrase list is impossible and statistical creation in unreliable.

E) Analysis of ADI Results

The results of the seven retrieval experiments are summarized in Figure 9. The plus or minus to the right of each figure indicates whether if is above (+) or below(-) the standard no-phrase value achieved for that recall level, (experiment 1). The results clearly show that there is no great gain achieved by the use of phrases and in some cases their

R	1	2	3	4	5	6	7
0.1	.6124	.6258+	.6500+	.5876-	.6124	.6333+	.5458-
0.2	.5524	.6258+	.6000+	.5276-	.5524	.6333+	.4858-
0.3	.4862	.4957+	.4798-	.4639-	.4826-	.5643+	.4862
0.4	.4287	.4078-	.4423+	.4223-	.4244-	.4392+	.4280-
0.5	.4335	.4059-	.4470+	.4096-	.4327-	.4309-	.4327-
0.6	.3351	.3234-	.3312-	.3208-	.3338+	.3441+	.3376+
0.7	.2608	.2742+	.2547-	.2608	.2617+	.2651+	.2586-
0.8	.2569	.2782+	.2426-	.2555-	.2571+	.2682+	2506-
0.9	.2493	.2675+	.2346-	.2433-	.2492-	.2590+	.2435-
1.0	.2493	.2675+	.2346-	.2433-	.2492-	.2590+	.2435-

1. = standard
2. = statistical, no occurrence
3. = syntactic, no cooccurrence
4. = statistical, cooccurrence
5. = syntactic, cooccurrence
6. = implicit, p=1
7. = relational

Summary of Phrase Method Results

Figure 9

use appears to be actually detrimental. However, upon more careful analysis of these results, a number of unusual factors are found which make these results somewhat less discouraging than they initially appear.

Consider first the results obtained with the statistical and syntactic phrases. It is argued in section C that the use of cooccurrence improves the retrieval quality. The results seem to indicate that exactly the opposite is true for experiment 4 and that experiment 5 results exceed experiment 3 at only half of the recall points. Upon analysis of the retrieval output it is discovered that the reason for this apparent turnabout is the dilution of nonrelevant concept vectors due to unmatched concepts. For many of the queries analyzed, there is one or more documents, highly correlated to that query, but nonrelevant, and which has a relatively large number of phrases which are not matched in the query. Because of the dilution effect which occurs when cooccurrence is not used, the correlations for these documents are lowered, often to a level below that of one of the relevant documents. The rank of the relevant document is thus raised by default even though its own correlation is not altered. Consider for example the correlation of document 11 with query A4. With no phrases used, this nonrelevant document ranks sixth with a correlation of 0.248189. The document has 13 statistical phrases which do not match the query. When retrieval is performed using these phrases without cooccurrence, the coefficient is reduced to 0.15599 and the rank lowered to ninth place. This allows one of the relevant documents to move ahead producing an apparent improvement in retrieval quality. When cooccurrence

is used there are no phrase matches, the coefficient remains 0.2418, and the relevant document is not allowed to move up. Considering the entire set of 33 documents relevant to the test queries, the ranks of 16 are improved by the use of statistical phrases with no cooccurrence. However, of these, only 7 actually move up in correlation coefficient. The remaining 9 lose in correlation but gain in rank due to the dilution and consequent lowering of nonrelevant documents. Ten of the 33 relevant documents lose in both rank and coefficient, mostly due to being diluted themselves, while 7 remained fixed in rank. Of these 7, 5 are reduced in coefficient but by an amount insufficient to drop the rank. Also most of the documents with a large number of phrases are not relevant to any test query. Thus the apparent superiority of the no-cooccurrence process (experiments 2 and 3) over the normal method (experiment 1) and the cooccurrence process (experiments 4 and 5) is almost entirely due to the lowering of the correlation coefficient of certain nonrelevant documents. This in turn is aided by the fact that most documents with a large number of phrases are not relevant to any query. The reduction in rank of these documents with respect to any query is thus guaranteed to cause, at worst, no harm and possibly produce a default raise in rank of a relevant document. This situation is clearly not typical. In general, every document must be considered as a potential relevant document. Lowering the rank for some set of documents for all queries would thus help retrieval in some cases, harm it in others. The results of experiments 2 and 3 reflect some positive effect caused by increasing the correlation in relevant documents. However, this effect is quite small. In general it can be concluded that since the conditions which led to the results of experiments 2 and 3 cannot be considered typical of document and query collections, the apparent improvement in

retrieval quality achieved with no-cooccurrence must therefore be held suspect.

Attention is next focused on experiments 4 and 5 which use statistical and syntactic phrases with the cooccurrence technique. When compared with experiment 1, the results seem to indicate that the cooccurrence processes are harmful to retrieval quality. However, this result is misleading as a result of a peculiar situation. This can be understood by considering the results of experiment 4. Of the 33 relevant documents, this phrase process improves both the rank and correlation for 9; 5 are reduced in rank; while the remaining 19 are unchanged. Overall this seems to be an improvement, but the tabulated results in Figure 9 do not bear this out. The reason for this lack of improvement lies almost entirely with query B5. It has only one relevant document and the phrase process lowers its rank from second to fifth thus lowering its precision for all recall levels from 0.5 to 0.2. This is a considerable decrease in precision, and since the values are averaged over only ten queries, the effect on the average is substantial. If precision values are taken for the nine other queries only, the values for the phrase processes exceed those for the no-phrase experiment for nearly all recall levels. Thus except for a rather unusual query, these phrase processes using cooccurrence provide some degree of improved retrieval results. The main drawback of such a process is the need for an a priori phrase list. And it is for this reason that the major emphasis in this study is on phrase methods which do not require predetermined lists.

The tabulations in Figure 9 indicate that results achieved by using the no-phrase-list method based on relational content analysis (experiment 7) are inferior to both the phrase list and no-phrase results. This is in

part due to the method's inability to associate phrases with different constituent concepts. The inferior results can also be blamed on the very small number of cooccurrences. Of the more than 800 relations entered, only 28 cooccurrences between documents and queries are found. This very low number can be blamed, at least in part, on the queries. They are all quite short and contain very few phrases. The queries also tend to be quite general. Since retrieval is performed by concept matching and not by hierarchical expansion, general queries do not always produce the desired results. Of the 28 cooccurrences, only 5 occur between a query and one of its relevant documents. In the ten test queries, three have no cooccurrences at all, and their results are clearly not altered from the no-phrase case. Four queries have cooccurrences in nonrelevant documents only and these results are obviously lowered. The three remaining queries have cooccurrences in relevant documents; however an improvement is realized in only one. Of the other two, one shows an improvement in correlation coefficient, but insufficient for a rank change, and the other has cooccurrences in nonrelevant documents which overshadow any improvement. These results might appear to cast some doubt on the value of this method. However this evidence is inconclusive and thus any decision is premature.

From the previous experiments it appears that the various phrase and structure methods can provide some degree of improvement in retrieval quality. But this improvement may be insufficient to warrant the additional work needed to use them. This deficiency, however, cannot be blamed entirely on weaknesses in the methods used. In the introduction to this study one of the primary uses of phrases in information retrieval is stated to be the separation of highly correlated, but not semantically identical, documents. A document collection must therefore contain such close documents in order

for phrases to demonstrate any significant retrieval improvement. To determine if the ADl collection provides a fair testbed for phrase use, a document-document correlation is preformed. The results indicate an average document-document correlation of 0.1 and a maximum of 0.8. This indicates that the ADI document space is in general quite sparse; but it may still contain some dense clumps of documents. To test for this, a third statistic is calculated; the average maximum document-document correlation (AMC). This is the correlation between a given document and its nearest neighbor averaged over all document-document pairs. In the ADI collection the AMC is less than 0.4 thus indicating the general absence of dense document clumps. Thus the documents in the ADI collection are seen to be quite spread out in the document space; and the extra dimension of refinement added to the documents and queries by the use of syntax is superfluous. Therefore to test more conclusively the usefulness of phrases in information retrieval, a more dense collection must be tried. Experiments with various other collections comprise the remainder of this study.

3. The Cranfield Collection

The Cranfield-424 Collection is a set of 424 documents in the field of aerodynamics. Because of its single specialized theme it might conceivably provide a denser collection on which to perform phrase experiments. Unfortunately this is not the case. Results of a document-document correlation are effectively the same as those for the ADI. The average document-document correlation is less than 0.1 and the AMC is about 0.4. It may therefore be expected that the Cranfield and ADI share the same undesirable characteristics concerning phrase use. For this reason the Cranfield collection is not used in this study.

4. The TIME Subset Collection

A) Construction

Because the existing collections do not exhibit the desired characteristics for conclusive testing of phrase techniques, a new collection is constructed. The process for creating such a collection is as follows. From an existing set of documents and queries, a subset of closely related queries is chosen. The set of documents relevant to any query in the subset is taken as the new document collection. The fact that these documents are all relevant to closely related queries guarantees that the documents themselves are also highly correlated. The collection chosen for this study is a set of articles from the "World" section of "TIME Magazine" (1963) with an associated set of current events queries. The largest number of related queries is six which deal with the Viet Nam war and particularly with the religious and political strife leading up to the overthrow of the Diem government. A total of 27 documents are relevant to these queries and this forms the TIME subset collection. The relatively small size of this document set detracts somewhat from the significance of the results of experiments using it, but not as much as might be expected. This is true for several reasons. First, the subset can be thought of as a single cluster in a large clustered document set. Since the subset contains all of the Viet Nam articles, its cluster centroid would clearly correlate highly with any Viet Nam related query. The real retrieval problem than becomes picking the desired articles from within the cluster. And second, the purpose of this set is to test the usefulness of phrases in information retireval, and phrases are micro rather than macro information retrieval aids. That is, the primary use for phrases is in determining fine differences in closely

related documents, and not in producing tremendous rank increases for low ranking documents. Thus this type of collection is sufficient for testing phrase processes.

The TIME articles are written in a very conversational and chatty style as opposed to the technical style fo the ADI and Cranfield collections For example, a document dealing with the Vietnamese coup begins:

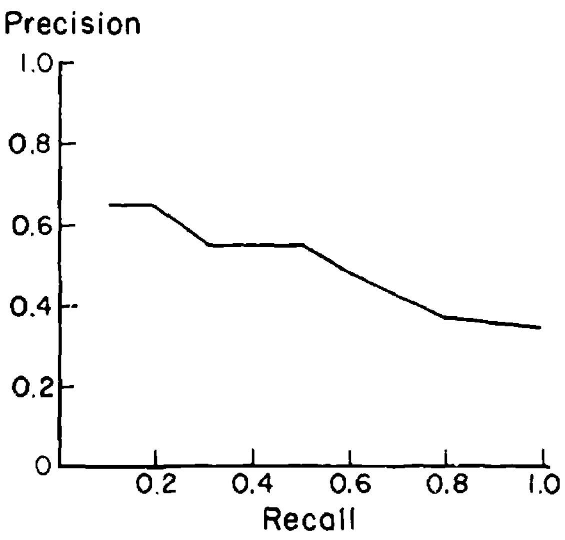
> Coping with Capricorn in business, count the costs before you act. The moon now in Capricorn suggests keeping practical values in mind. Tomorrow is rather too energetic for comfort, but that may be because everybody is on the move. (A late August horoscope.) Syndicated horoscopes, many of them from abroad are a popular feature in many South Vietnamese newspapers, but last week the government banned them, presumably on a theory that some star-minded dissident might be moved to try a coup on an astrologically auspicious day.
> ["TIME", 9/6/63, page 19]

The article then presents its true purpose, that of describing the increasing United States dissatisfaction with the present South Vietnamese government and the possibility of an American-encouraged coup. The article goes on by describing the martial law measures being taken by the Vietnamese government to prevent a coup, and then gives a brief biography of several generals who might stage the coup. Thus the crux of the article is to describe the tenuous political situation in Viet Nam, not to discuss astrology. The paragraph quoted above thus serves merely as a light introduction.

Construction of document vectors from the full text of articles such as this could very well result in a tremendous amount of spurious information in the vector. For this reason, and because of the document length, it is necessary to form abstracts. The abstracts used are about one hundred words

in length and present the main ideas of the article using much the same vocabulary and constructions as in the original text. The abstracts thus capture the gist of the article in both content and style while eliminating most of the unrelated chaff. Using these abstracts, a vocabulary is constructed and document vectors are formed using standard SMART dictionary construction and vector creation programs. The dictionary assigns a single concept number to all words with a common stem. Figure 10 presents the results of a normal SMART search with the TIME subset collection. The results are consistent with retrieval results using other collections. There thus seems to be nothing particularly unusual about this document and query set which might tend to diminish the significance of any experimental results.

Three sets of phrase experiments are performed using the TIME subset collections. The first two are the implicit and relational as presented earlier. As before, various parameters are used to weight the importance of a phrase match in the correlation calculation. A third phrase process called <u>half relational</u> is also used. This is a weaker form of relational phrase matching (heretofore referred to as full relational for clarity). In full relational, a phrase match occurs only when the document and query have the same concept pair and the concepts are joined by the same relation. In Figure 11 below, the query phrase QP matches only document phrase DP1. In half relational matching, a match occurs when the document and query share a concept which occurs in the same relational context in both vectors. For example in Figure 11, the query QP matches document phrases DP1, 2, and 3 but not 4. While the query concept matches in DP4, the relational context does not. That is, in QP concept 5 is a modifier while in DP4 it is modified. Thus as the name implies, half relational matches require only one of the two related concepts to match. This is clearly a weaker matching [illegible]ment and is expected to produce more matches than full relational. This



Standard Re^ults, Time Subset

Figure 10

could be of value in cases where cooccurrences of whole phrases are rare; but it may also give many improper matches.

```
QP  < 5, 7,MOD>

DP1 < 5, 7,MOD>
DP2 < 5, 9,MOD>
DP3 <13, 7,MOD>
DP4 < 3, 5,MOD>
```

Sample Query and Document Relations Phrases

Figure 11

The results for these experiments are shown in Figure 12 A, B, and C. Figure 13 gives the tabulated results for each method using the weighting parameter which provides the best results. While these represent the best values, the results achieved for other parameter values are only very slightly lower. As before the figure shows whether the results of the phrase experiment are above (+) or below (-) those achieved when no phrases are used. These results reveal that implicit phrase matching is harmful to retrieval quality and gets worse as the weighting parameter is increased. Half relational shows some slight improvement for low recall values while full relational is generally worse. However in these latter two methods, all differences are very small and effectively insignificant.

B) Analysis of Results

The most surprising result of this set of experiments is the harmful effect caused by implicit phrases. This is inconsistent with the results obtained with the ADI collection. This apparent turnabout can be explained by recalling the original purpose for using implicit phrases. This is to separate those documents whose correlation is based on a cooccurrence of

RECALL	STANDARD	TIME IMPLICIT PHRASES P = 0.5	P = 1.0	P = 1.5	P = 2.0
0.1	.6426	.6333-	.5639-	.5635-	.5635-
0.2	.6426	.6333-	.5639-	.5635-	.5635-
0.3	.5537	.5778+	.5639+	.5635+	.5635+
0.4	.5500	.5361-	.5125-	.5135-	.5135-
0.5	.5500	.5361-	.5125-	.5135-	.5135-
0.6	.4781	.4604-	.4447-	.4429-	.4429-
0.7	.4217	.4215-	.4256+	.4183-	.4183-
0.8	.3745	.3652-	.3564-	.3579-	.3579-
0.9	.3702	.3577-	.3555-	.3496-	.3496-
1.0	.3669	.3577-	.3555-	.3496-	.3496-

Summary of TIME Implicit Phrase Experiments

Figure 12A

RECALL	STANDARD	TIME FULL RELATIONAL PHRASES P = 0.5	P = 1.0	P = 1.5	P = 2.0
0.1	.6426	.6389-	.6359-	.6333-	.6333-
0.2	.6426	.6389-	.6359-	.6333-	.6333-
0.3	.5537	.5500-	.5803+	.5778+	.5778+
0.4	.5500	.5417-	.5215-	.5190-	.5190-
0.5	.5500	.5417-	.5215-	.5190-	.5190-
0.6	.4781	.4614-	.4520	.4578-	.4634-
0.7	.4217	.4079-	.4041-	.4099-	.4154-
0.8	.3745	.3632-	.3602-	.3577-	.3577-
0.9	.3702	.3632-	.3602-	.3577-	.3577-
1.0	.3669	.3632-	.3602-	.3577-	.3577-

Summary of TIME Full Relational Phrase Experiments

Figure 12B

RECALL	STANDARD	TIME HALF RELATIONAL PHRASES P = 0.5	P = 1.0	P = 1.5	P = 2.0
0.1	.6426	.6274-	.6274-	.6274-	.6663+
0.2	.6426	.6274-	.6274-	.5857-	.6107-
0.3	.5537	.5218-	.5163-	.5718+	.6107+
0.4	.5500	.5218-	.5112-	.5649+	.5788+
0.5	.5500	.5218-	.5112-	.5649+	.5788+
0.6	.4781	.4448	.4362-	.4468-	.4468-
0.7	.4217	.4111-	.4062-	.4111-	.4111-
0.8	.3745	.3395-	.3350-	.3259-	.3198-
0.9	.3702	.3395-	.3350-	.3259-	.3198-
1.0	.3669	.3372-	.3327-	.3236-	.3175-

Summary of TIME Half Relational Phrase Experiments

Figure 12C

RECALL	STANDARD	IMPLICIT P= 0.5	FULL P= 0.5	HALF P = 2.0
0.1	.6426	.6333-	.6389-	.6333+
0.2	.6426	.6333-	.6389-	.6107-
0.3	.5537	.5778+	.5500-	.6107+
0.4	.5500	.5361-	.5417-	.5788+
0.5	.5500	.5361-	.5417-	.5788+
0.6	.4781	.4604-	.4614-	.4468-
0.7	.4217	.4215-	.4079-	.4111-
0.8	.3745	.4652-	.3632-	.3198-
0.9	.3702	.3577-	.3632-	.3198-
1.0	.3669	.3577-	.3632-	.3175-

Summary of TIME Processes

Best Results Used for Each

Figure 13

several concepts in the document and query from those docume· ·s whose correlation results from one or two highly weighted concepts. In the ADI collection, there are many concepts in the documents with weights of twenty-four or more so that there is a real need for such a separa·ion technique. As a result, implicit phrases provide improved retrieval for the ADI. In the TIME collection occurrences of highly weighted concepts are much rarer than in the ADI. Consequently the reason for using implicit phrases does not exist. Employing the phrase technique thus does not accomplish the purpose for which it is designed and hence no improvement is realized. Thus it appears that implicit phrases may be a useful technique but only when used with collections which meet certain requirements as to the presence of highly weighted concepts.

The results achieved using both half and full relational content analysis are discouraging. They may be the result of weakness in the phrase process or, as in the case of the ADI collection, they may be caused by the collection itself. Figure 14 shows for each method how many phrases are matched with relevant and nonrelevant documents. In both cases only about one-third of the phrase matches are between a query and one of the relevant documents. This seems to indicate that the weakness may lie in the phrase matching method, however this is only partially true. The reason for the poor results for the half relational is simply that the matching criteria are too weak. Too many false and incorrect phrases are matched and the lower retrieval quality results. It therefore seems the half relational method is worthless although some further testing is necessary to finalize the decision. The reason for the poor results with the full relational method is not so clearly the fault of the matching scheme. Of the 82 phrase

matches between documents and queries, 65, or roughly 80%, are matches of the phrases "South Viet" or "Viet Nam". Since the entire collection deals with South Viet Nam, these phrases occur almost uniformly throughout the document set. And since each query has an average of three times as many nonrelevant as relevant documents, the results in Figure 14 are to be expected. If this document collection were considered as one cluster of a larger collection, the phrase South Viet Nam would be useful in gaining access to the cluster. However, within the cluster it is a poor discriminator and thus cannot help retrieval. If South Viet Nam is removed from the set of phrase matches, more than two-thirds of the remaining phrase matches occur between a query and a relevant document and retrieval would clearly be improved. However the small number of relations that remain seem to indicate the same collection sparseness as is found in the ADI and Cranfield collections.

	Number of Phrase Matches				
	With Rel Documents		With Nonrel Documents		Total
Half Relational	89	32%	186	67%	277
Full Relational	28	34%	54	65%	82

Phrase Matches (TIME)

Figure 14

A document-document correlation on the TIME subset collection reveals that the average correlation is 0.2. This is twice as high as the ADI or anfield and is to be expected since the TIME collection is designed

specifically for high density. However, the average maximum correlation (AMC) which is a more important measure is 0.41, roughly the same as for previous collections. This indicates that the increased density in the collection is achieved by the omission of low correlating documents, and not by the occurrence of highly correlated document pairs. And this collection is seen as no better for phrase experimentation than the ADI. Thus is appears that even though this collection is constructed specifically for phrase use, it does not satisfy some of the theoretical prerequisites. The natural question at this point is exactly in what type of collection are phrases useful. This question is treated in the next section.

Beside collection density, there is another factor affecting the usefulness of phrases. This is the type of relations occurring between text elements. There are basically two types of semantic relations by which phrase words may be associated: reversible and nonreversible. A reversible relation is one in which the ordering of the constituent words has no effect on the meaning. For example the words "information" and "retrieval", occurring in almost any structure means "information retrieval", and hence the words are related by a reversible relation. A nonreversible relation is one in which the phrase structure is significant. The relation between "U. S." and "Russia" in the sentence below is an example of a nonreversible relation.

The U. S. influences Russia.

There is also a third type of relation, which is usually a specialized subset of nonreversible, called trivial nonreversible. These are phrases whose meaning depends on the structure and are technically nonreversible.

However, with these special phrases, all but one of the potential meanings do not occur in practice, and the relation assumes reversible characteristics. For example, consider the sentence:

The U. S. invades Cambodia.

Since it is possible for the U. S. to invade Cambodia and vice versa, the relation between U. S. and Cambodia is clearly nonreversible. However, since in fact Cambodia has not and probably will never invade the United States, the relation is actually trivial nonreversible and hence its structure becomes unimportant. As mentioned earlier, one of the primary objectives of the use of structured phrases is in matching phrases whose meaning is a function of both its content and its structure, that is, phrases with nonreversible relations. If such phrases do not occur in the analyzed text, structured phrase use can clearly provide little or no help in retrieval. This is the case in the TIME collection. Of the phrases isolated, a vast majority are reversible or trivial nonreversible. Thus the lack of nonreversible relations is another reason for the failure of the content analysis scheme to achieve improved results.

5. A Third Collection

In the previous sections it is shown that the ADI and TIME collections do not require the use of phrases because they do not demonstrate the characteristics which provide the theoretical basis of phrase use. They are neither dense enough nor do they contain large numbers of nonreversible relations. And hence no significant advantage is gained through the use of phrases. Analysis of other natural collections such as the Cranfield reveals the same situation. The natural question at

this point is this: what is a collection like which has the desired characteristics? To attempt to answer this a purely artificial collection is constructed. The collection consists of twenty documents and fourteen queries, each in the form of a short sentence. The subject matter deals with the relation between birds and worms and is inspired by an example by Simmons [8]. This highly specific subject guarantees a highly dense document space. In addition, the documents are specifically written to include nonreversible relations. For example, in

Birds eat worms.

Worms eat grass.

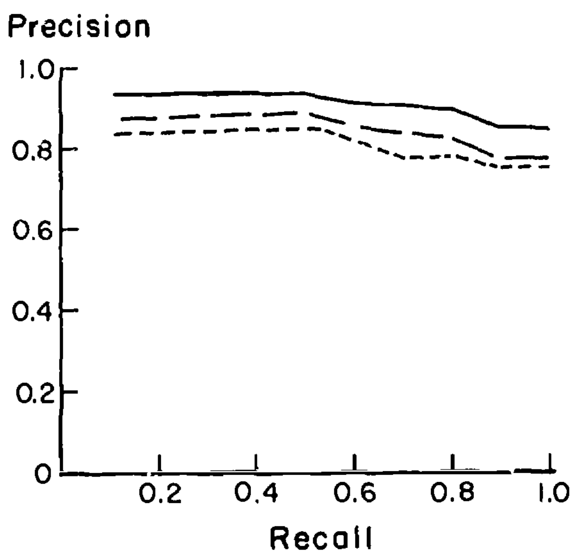
The words "worms" and "grass" are clearly nonreversibly related. This collection might thus be considered an ideal testbed for phrase experimentation.

Results are tabulated in Figure 15 and shown graphically in Figure 16. Because of the extreme closeness of the various results, only the best of each set is shown. Also the results of implicit phrases are not shown on the graph in Figure 16 since they coincide with the no phrase results. The lack of improvement here is caused, as in the TIME collection, by the lack of highly weighted concepts in the document and query vectors. Thus the problem which implicit phrases are designed to solve simply does not exist. The results for half relational phrases show a slight improvement at all recall levels. More important, however, are the results in Figure 17. This indicates that only about a third of the half relational phrase matches are between a query and one of its related documents. This seems to finalize the conjecture stated earlier that half relational matching is too weak a criterion and results in too many improper phrase matches.

RECALL	STANDARD	IMPLICIT	FULL	HALF
0.1	.8440	.8440	.9286+	.8810-
0.2	.8440	.8440	.9286+	.8810-
0.3	.8440	.8440	.9286+	.9810-
0.4	.8440	.8440	.9286+	.8810-
0.5	.8440	.8440	.9286+	.8810-
0.6	.8083	.8383	.9000+	.8524+
0.7	.7798	.7798	.9000+	.8524+
0.8	.7798	.7798	.8929+	.8333+
0.9	.7548	.7548	.8393+	.7554+
1.0	.7548	.7548	.8393+	.7554+

Summary of B&W Phrase Processes

Figure 15



---- implicit (coincides with standard)

—— full relational

—— half relational

---- standard

B & W Phrase Results

Figure 16

It thus appears to be an unsuitable phrase process. As Figure 17 indicates, quite the opposite is true for full relational phrases. More than two-thirds of the full relational phrase matches are with relevant documents. This fact is also reflected in the improved precision at all recall levels achieved by any full relational matching. These results can be treated both optimistically and pessimistically. On the one hand, they show conclusively that structural phrases can be of value in information retrieval. On the other hand, this improvement in retrieval results is not achieved in "natural" collections such as the ADI, but rather only for one which is highly artificial and contrived. It is not clear at this point whether any natural collection can meet all of the requirements for advantageous phrase use.

6. Conclusion

The general conclusions that can be drawn from these experiments are that a number of different types of phrase processes are useful in information retrieval provided certain characteristics exist in the document set. This is especially true in the case of structural phrases where it appears that effective phrase use depends more on the collection than on the specifcc phrase process.

The implicit phrase process is designed to boost correlations based on the cooccurrence of many concepts in the document and query as opposed to those correlations which are the result of a very few matches of highly weighted concepts. Results indicate that it performs the job quite well. However, if the collection has relatively few high weights, the need for implicit phrases no longer exists. Using implicit phrases with such collections is thus a wasted effort and may even lead to downgraded retrieval quality.

	NUMBER OF PHRASE MATCHES				
	WITH REL DOCUMENTS		WITH NONREL DOCUMENTS		TOTAL
HALF RELATIONAL	62	38%	102	62%	164
FULL RELATIONAL	36	69%	16	31%	52

Phrase Matches (B & W)

Figure 17

For structured phrases to be of value in information retrieval, a number of conditions must be met. First the collection must be sufficiently dense, or at least have some dense clumps of documents. Second, the document must contain nonreversible relations. Along the same line, the documents in any particular clump must be sufficiently different semantically so that conceivably some but not all could be relevant to a given query. In other words, there must be a potential need to discriminate between closely related documents. This restriction is necessary for the following reason. It is conceivable that a particular clump of documents could be so closely related that either all or none are related to any query. While this clump satisfies the density requirement and may have nonreversible relations as well, it does not require the use of phrases. There is no need to distinguish among members of the clump and thus phrases cannot help. Finally, it is necessary that the queries contain nonreversible relations. If such relations are not requested in the query, as is true in the ADI collection, no advantage is gained by using them in the documents. Testing this condition is easy when dealing with experimental documents and queries, but clearly impossible in real applications. However, it is possible to predict the general form for expected queries and thereby determine if they meet the phrase requirement. As a general guideline, queries are more applicable to phrase use if they are of the question-answering variety rather than pure document retireval.

The final conclusion that is reached from this study is that, contrary to intuition, phrases do not seem to exert a large effect on a user choice of relevant documents. Future work must be done on determining the factors that go into a user's relevancy decisions. With more insight into this area, the role of structure in information retrieval will become much more clearly defined.

References

[1] Curtice, R. M., and Jones, P. E., An Operational Interactive Retrieval System, Arthur D. Little, Inc., 1969.

[2] Douglas, E., Mandersloot, W., and Spicer, N., Thesaurus Control — the Selection, Grouping, and Cross-referencing on Terms for Inclusion in a Coordinate Index Word List., Journal of the American Society for Information Science, January-February, 1970.

[3] Hutchins, W. J., Automatic Document Selection Without Indexing, Journal of Documentation, Vol. 23, No. 4, December 1967.

[4] IBM Systems/360 Document Processing System, Applications Description, IBM, 1967.

[5] Lesk, M. E., A Proposal for English Text Analysis, Bell Telephone Laboratories, 1969.

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

[7] Salton, G., Automatic Text Analysis, Science, Vol. 168, April 17, 1970.

[8] Simmons, R. F., Synthex. In Orr, W. D., (Ed.), Conversational Computers, John Wiley and Sons, Inc., New York, 1968.

[9] Weiss, S. F., A Template Approach to Natural Language Analysis for Information Retrieval, Ph.D. Thesis, Cornell University, Ithaca, New York, 1970.

II. The "Generality" Effect and the Retrieval Evaluation for Large Collections

G. Salton

Abstract

The retrieval effectiveness of large document collections is normally assessed by using small subsections of the file for test purposes, and extrapolating the data upward to represent the results for the full collection. The accuracy of such an extrapolation unhappily depends on the "generality" of the respective collections.

In the present study the role of the generality effect in retrieval system evaluation is assessed, and evaluation results are given for the comparison of several document collections of distinct size and generality in the areas of documentation and aerodynamics.

1. Introduction

Over the past few years a great many studies have been undertaken in an attempt to assess the retrieval effectiveness of a variety of automatic analysis and search procedures. Under normal circumstances, a single test collection is used which is subjected to a variety of processing methods; paired comparisons are then made between two or more procedures for this collection in order to determine which methods are most effective in a retrieval environment. [1,2,3]

Occasionally, however, it is necessary to use several different

document collections in a test situation and to compare the results for distinct collections (rather than for distinct processing methods). Such is the case notably when a variable is tested for which a single collection is not normally usable (for example, the language in which the documents are written [4]), or when an attempt is made to extrapolate from a small test collection to a large operational one. [5] In such situations, special precautions are needed to insure that the evaluation measures actually reflect the performance differences between the respective collections.

Consider as an example, two distinct document collections. Performance differences might then emerge as a result of the following collection characteristics:

a) differences in subject matter;

b) differences in the scope of the collections;

c) differences in the document types available for processing;

d) differences in query types;

e) differences in the collection size;

and f) differences in the relevance judgments of queries with respect to documents.

In the present study, the first four variables are not under investigation in the sense that comparisons are made only for collections of document abstracts of similar scope within a specific subject area, using standard user requests of the type often submitted to an information center. The other two variables, namely collection size and type of relevance assessments are of special interest, since both of them affect the evaluation results obtained for large operational systems. These variables to a large

extent determine the generality of the collection, that is, the average number of relevant items per query, and generality in turn affects the evaluation parameters.

In the remainder of this study, two different generality problems are examined by using on the one hand collections of different size for which the relevance judgments agree, and, on the other hand, collections of identical size with different relevance properties. The variations obtained in the evaluation results are examined, and an attempt is made to interpret the respective performance differences.

2. Basic System Parameters

The evaluation parameters used to assess the retrieval performance of a given set of user queries with respect to a document collection are normally based on a two by two contingency table which distinguishes between the documents retrieved in answer to a given query and those not retrieved, and between items judged to be relevant to the query and those not relevant. A typical contingency table is presented in Table 1(a), and four common evaluation measures derived from it are contained in Table 1(b).

Each of the measures listed in Table 1 is initially defined for each query separately. However, procedures exist for averaging the measures over a complete query set and for suitably displaying the resulting values in the form of recall-precision, or recall-fallout graphs. [6] These graphs are then expected to reflect the performance of an entire system for a given set of users.

It should be noted that the four retrieval measures are not

	Relevant	Not Relevant	
Retrieved	a	b	a+b
Not Retrieved	c	d	c+d
	a+c	b+d	a+b+c+d

(a) Contingency Table

Symbol	Evaluation Measure	Formula	Explanation
R	Recall	$\frac{a}{a+c}$	proportion of relevant actually retrieved
P	Precision	$\frac{a}{a+b}$	proportion of retrieved actually relevant
F	Fallout	$\frac{b}{b+d}$	proportion of nonrelevant actually retrieved
G	Generality	$\frac{a+c}{a+b+c+d}$	proportion of relevant per query

(b) Principal Evaluation Measures

Retrieval Evaluation Measures

Table 1

independent of each other. Specifically, three of the measures will automatically determine the fourth. As an example, equation (1) can be used to derive precision in terms of recall, fallout, and generality, as follows:

$$P = \frac{R \cdot G}{(R \cdot G) + F(1-G)} \qquad (1)$$

Most of the retrieval evaluation results published in the literature have been presented in terms of recall and precision. Since recall provides an indication of the proportion of relevant actually obtained as a result of a search, while precision is a measure of the efficiency with which these relevant are retrieved, a recall-precision output is user-oriented, in the sense that the user is normally interested in optimizing the retrieval of relevant items. On the other hand, fallout is a measure of the efficiency of rejecting the nonrelevant items, and includes as a factor the total number of nonrelevant in the collection (which in many cases is approximately equivalent to the collection size). For this reason, a recall-fallout display is normally considered to be systems-oriented since it indicates how well the nonrelevant are rejected as a function of collection size.

In view of their special orientation, it would then appear that some of the measures are more appropriate in certain circumstances than in others: in particular, if a systems viewpoint is important which takes into account the amount of work devoted to the retrieval of nonrelevant items as well as the collection size, a fallout display may be more desirable than a graph based on precision.

The situation is unfortunately complicated by the fact that the

various measures do not vary in the same manner when a comparison is made of the performance of several distinct document collections. Consider, as as example, the parameter variations produced by changes in collection generality. As the generality increases, that is, as the average number of relevant per query grows larger, the number of relevant retrieved may also be expected to increase. In terms of the variables introduced in Table 1, a and a+c may then be expected to grow directly with generality; on the other hand a+b, and b+d (the total retrieved, and the total non-relevant) remain relatively constant.

As G increases, the following picture then emerges for R, P, and F, respectively:

$$R = \frac{\uparrow}{\uparrow}, \quad P = \frac{\uparrow}{\rightarrow}, \quad F = \frac{\rightarrow}{\rightarrow}$$

where the upward arrow denotes an increasing quantity, and the horizontal arrow a quantity more or less constant. Thus, R and F should remain reasonably constant with changes in generality, since numerator and denominator vary in the same direction. Precision, on the other hand, should vary directly with generality because of the increasing numerator together with the constant denominator.

This kind of argument has been used in the past to show that the use of recall-precision graphs is generally undesirable, [7], and to claim that performance figures obtained with small sample collections in a laboratory environment cannot be applied to large operational collections [8]. This question is further examined in the next section.

3. Variations in Collection Size

A) Theoretical Considerations

Consider a performance comparison for two collections of different size within a given subject area. Such collections generally exhibit different generality characteristics, since the larger collection is likely to contain on the average many more nonrelevant items per query, and therefore proportionately many fewer relevant ones.

In going from the smaller (test) collection to the larger (operational) one, two limiting cases may be distinguished:

a) if the relevance of the documents added to the small collection in order to produce the large one is difficult to assess in a clear-cut way, and nonrelevant items that are hard or easy to reject are added roughly in the same proportion as originally present, then for a given level of recall a larger number of relevant items will have to be retrieved; this will imply the simultaneous retrieval of a larger number of nonrelevant, thereby depressing precision, but keeping fallout roughly constant;

b) on the other hand, if the documents added are clearly extraneous to the query topics and the nonrelevant ones are easily rejectable, the number of relevant and nonrelevant retrieved at a given recall level remains constant, thereby producing a constant precision but lower fallout for the larger collection: the situation is summarized in Table 2.

If case 2 were to occur in practice, that is, if one could insure that any nonrelevant documents added to the small collection would be

	Large Collection	
Small Collection	(1) Addition of Partly Relevant and Non-relevant in same Proportion	(2) Addition of Extraneous Clearly Non-relevant
P F	P $\downarrow$ F $\rightarrow$	P $\rightarrow$ F $\downarrow$

Precision and Fallout Performance for Variations in Collection Size

Table 2

easy to reject, then the standard recall-precision plot would furnish a completely adequate evaluation tool, since the precision would then be independent of the generality change, and would in fact be identical for both collections at each common recall level. If, on the other hand, case 1 is taken as typical, then fallout can be assumed to be constant. This makes it possible to compute an "adjusted precision" value as a function of generality, to account for the generality change in upgrading from a small collection to a large one.

Consider, as an example, a document collection with generality G_1, and a given precision P_1 at a recall level of R_1. If the size of the collection is altered to a new generality G_2, then, for any given recall level, equation (1) can be used to compute the adjusted precision P_2 for the larger collection. In fact, if the generality change is subject to the rules of case 1, one has (from equation (1)):

$$P_2 \text{ (adjusted)} = \frac{R_1 \cdot G_2}{(R_1 \cdot G_2) + F_1(1-G_2)} \qquad (2)$$

where the computations are made for a given recall level $R_1 = R_2$, and fallout is assumed constant. Equation (2) then provides a means for computing the precision transformation for the case where all factors other than generality remain constant.

Cleverdon and Keen propose a three-step procedure for effecting the precision transformation as follows: [1]

a) given G_1, R_1 and P_1 compute F_1;

b) assume $F_1 = F_2$;

c) given G_2, $R_1 = R_2$, and F_1, compute P_2.

An example for a collection of generality 0.005 and recall and precision values of 0.60 and 0.25 respectively is shown in Table 3. The precision adjusted to a generality level of G = 0.002 is seen to be 0.11.

B) Evaluation Results

The theoretical considerations outlined in the last few paragraphs indicate that the retrieval evaluation provides an accurate picture for the case where the expansion in collection size is caused by the addition to a small document collection of clearly nonrelevant items which are easily rejectable, and for the case where fallout remains constant, that is, where relevant and nonrelevant items are added in a proportion roughly equivalent to that which originally existed.

Unfortunately, when the assumptions of cases 1 and 2 are tested on actual document collections of different generality, they are found not to hold in practice. For example, in a test conducted some years ago with two document collections of 200 and 1400 documents in aerodynamics, respectively, and a sample of 42 queries, Cleverdon and Keen found for a specified cutoff and processing method that

> "b (the nonrelevant retrieved) has increased by a factor of 5.2352 while the total number of nonrelevant documents in the collection (b+d) has increased by a factor of 7.1443." [1, p.74]

For the example considered, fallout therefore did not remain constant, and many of the nonrelevant included in the larger collection of 1400 items obviously exhibited a lower probability of being retrieved than the nonrelevant included in the smaller subcollection.

	Large Collection	
Small Collection	1 Addition of Partly Relevant and Nonrelevant in same Proportion	2 Addition of Extraneous Clearly Nonrelevant
P	P ↓	P →
F	F →	F ↓

Precision and Fallout Performance for Variations in Collection Size

Table 2

Parameter		Collection 1		Collection 2
G		.005		.002
R		.60		.60
P	step 1	.25	step 3	.11 (adjusted P)
F		.00905		.00905
			step 2	

Precision Transformation for Constant Fallout

Table 3

To verify this result, the two collections originally used by Cleverdon were subjected to a complete retrieval test, using a set of 36 queries with identical relevance properties in both collections (the set of relevant items was the same for each query in both collections). The collection characteristics are summarized in Table 4, and recall-precision, as well as recall-fallout, plots are included in Fig. 1, averaged over the 36 test queries. [9]

It may be seen from the output of Table 4 and Fig. 1 that although the collection generality decreases by a factor of about seven in the transition from small to large collection, the fallout decreases by a factor of only three on the average. Thus the proportion of nonrelevant retrieved is much smaller for the large collection than for the small one, producing the recall-fallout plot of Fig. 1(b) which favors the large collection (the smaller the fallout, the better is the performance).* The recall-precision plot, on the other hand, favors the small collection (the higher the precision, the better is the performance), indicating that at a given recall level, fewer nonrelevant will have been retrieved for the small collection than for the large one.

The data of Table 5, containing the average number of nonrelevant documents retrieved at various recall levels, indicate that the seven-fold decrease in collection generality is accompanied by an increase in the average number of nonrelevant retrieved, ranging from a factor of 2 at a recall of 0.[illegible], a factor of only 3.2 at a recall of 0.3 and 0.5. This explains the superior systems-oriented performance of the large 1400 collection in comparison with the small one.

---

*The average number of nonrelevant items retrieved at various recall levels shown in Table 5 for the Cranfield 200 and 1400 collections.

Property	Cranfield 200	Cranfield 1400
Source	Cranfield document abstracts in aerodynamics	Cranfield document abstracts in aerodynamics
Document Analysis	Word stem process	Word stem process
Number of Documents	200	1400
Number of Queries	36	36
Number of Relevant Documents	160	160
Type of Search	Full search	Full search
Generality	.0222	.0031
Average Fallout	.0248	.0081

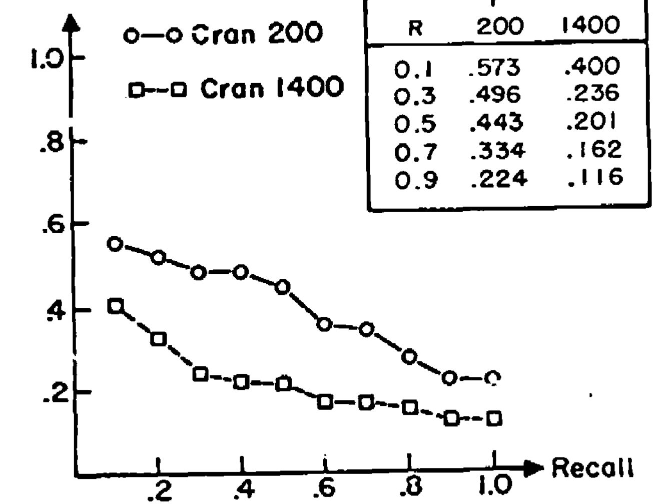
Collection Properties for Cranfield 200 and 1400

Table 4

Recall	Average Number of Nonrelevant Retrieved		Factor of Increase from 200 to 1400
	Cranfield 200	Cranfield 1400	
0.1	0.33	0.67	2
0.3	1.35	4.32	3.2
0.5	2.79	8.82	3.2
0.7	6.21	16.15	2.6
0.9	13.89	30.54	2.2

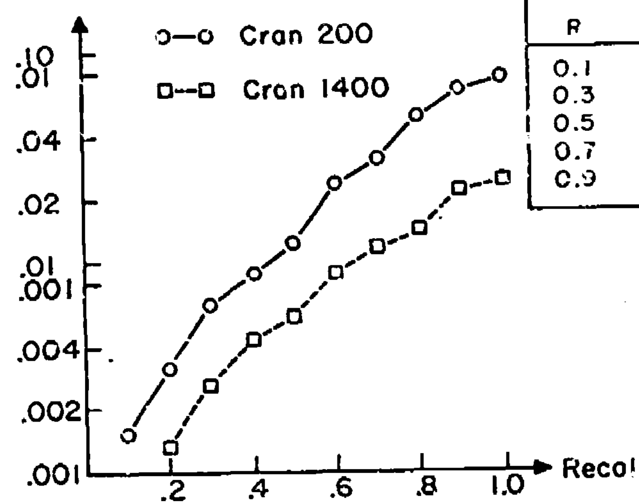
Increase in Nonrelevant Retrieved
from Cranfield 200 to Cranfield 1400

Table 5



R	P 200	P 1400
0.1	.573	.400
0.3	.496	.236
0.5	.443	.201
0.7	.334	.162
0.9	.224	.116

a) Recall-Precision



R	F 200	F 1400
0.1	.0015	.0002
0.3	.0066	.0029
0.5	.0139	.0059
0.7	.0314	.0111
0.9	.0706	.0207

b) Recall-Fallout

Performance Comparison for Cran 200 and Cran 1400 Collections
(averages over 36 queries; word stem process)

Fig. 1

In practice, it is seen that the larger the collection (and therefore the smaller the generality), the larger will be the number of nonrelevant items which will have been retrieved at any given recall level; however the resulting decrease in precision performance is much smaller than expected by the factor of increase in collection size and nonrelevant items added. Neither of the two simple generality transformations discussed in the preceding subsection appears to be applicable in practice, since both precision and fallout may be expected to decrease with a decrease in collection generality.*

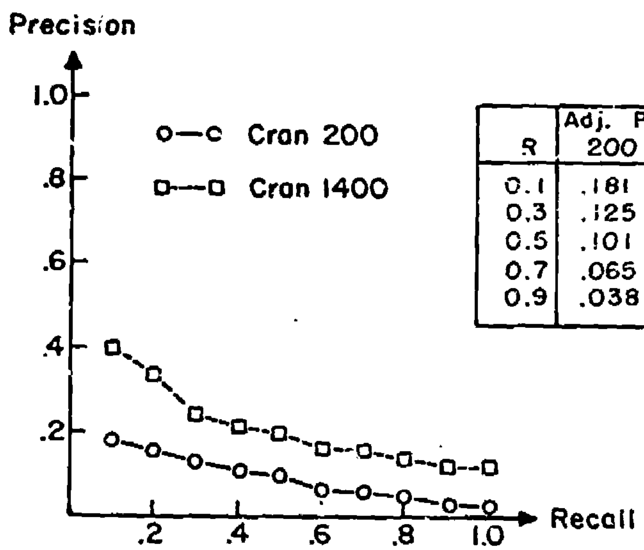
C) Feedback Performance

It is known that interactive search methods in which the user influences the retrieval process by providing appropriate feedback information during the course of the operations can be used profitably in a retrieval environment. [10,11] In fact, some of the feedback methods which have been tested over the last few years, including, in particular, the <u>relevance feedback</u> process regularly used with the automatic SMART document retrieval system, provide anywhere from five to twenty percent improvement in precision at a given recall level. Most other refinements in retrieval methodology — such as, for example, a particularly sophisticated language analysis scheme — may bring improvements in performance of the order of a few percent at best.

The relevance feedback process utilizes user relevance judgments

---

*If the precision transformation of equation (1) were (incorrectly) to be applied to the precision performance of the small collection to reduce its generality to that of the large collection (.0031), the adjusted precision curve of Fig. 2 would result. This adjusted precision is an inverse function of fallout, which accounts for its inferior performance compared with that of the large collection.



R	Adj. P 200	P 1400
0.1	.181	.400
0.3	.125	.236
0.5	.101	.201
0.7	.065	.162
0.9	.038	.116

Recall–Precision Plot for Cran 200 and Cran 1400 Collections
(Precision Adjusted to Generality of .0031)
(averages over 36 queries)

Fig. 2

for documents previously retrieved by an initial search in order to construct an improved query formulation which can subsequently be used in a new "first iteration", or "second iteration" search. Specifically, an initial search is performed for each request received, and a small amount of output, consisting of some of the highest scoring documents, is presented to the user. Some of the retrieved output is then examined by the user who identifies each document as being either relevant (R) or not relevant (N) to his purpose. These relevance judgments are later returned to the system, and used automatically to adjust the initial search request in such a way that query terms present in the relevant documents are promoted (by increasing their weight), whereas terms occurring in the documents designated as nonrelevant are similarly demoted. This process produces an altered search request which may be expected to exhibit greater similarity with the relevant document subset, and greater dissimilarity with the nonrelevant set.

The altered request can next be submitted to the system, and a second search can be performed using the new request formulation. If the system performs as expected, additional relevant material may then be retrieved, or, in any case, the relevant items may produce a greater similarity with the altered request that with the orig[illegible]. The newly retrieved items can again be examined by the user, ar[illegible]vance assessments can be used to obtain a second refoumulation [illegible]e request. This process can be continued over several iterations, until such time as the user is satisfied with the results obtained.

In order to determine whether the relevance feedback process is usable with large document collections in an operational environment, the feedback procedure was tested using two collections in aerodynamics of different generality. [12] If comparable feedback improvements could be obtained for collections of varying size and generality, then it would appear reasonable to conclude that the feedback process will be valuable under operational conditions.

The two collections being tested consist of 200 and 424 document abstracts in aerodynamics, respectively, together with 22 queries with identical relevance properties in both collections. The collection characteristics are summarized in Table 6, and the recall-precision and recall-fallout graphs obtained with a "positive" feedback strategy are shown for both collections if Figs. 3 and 4.

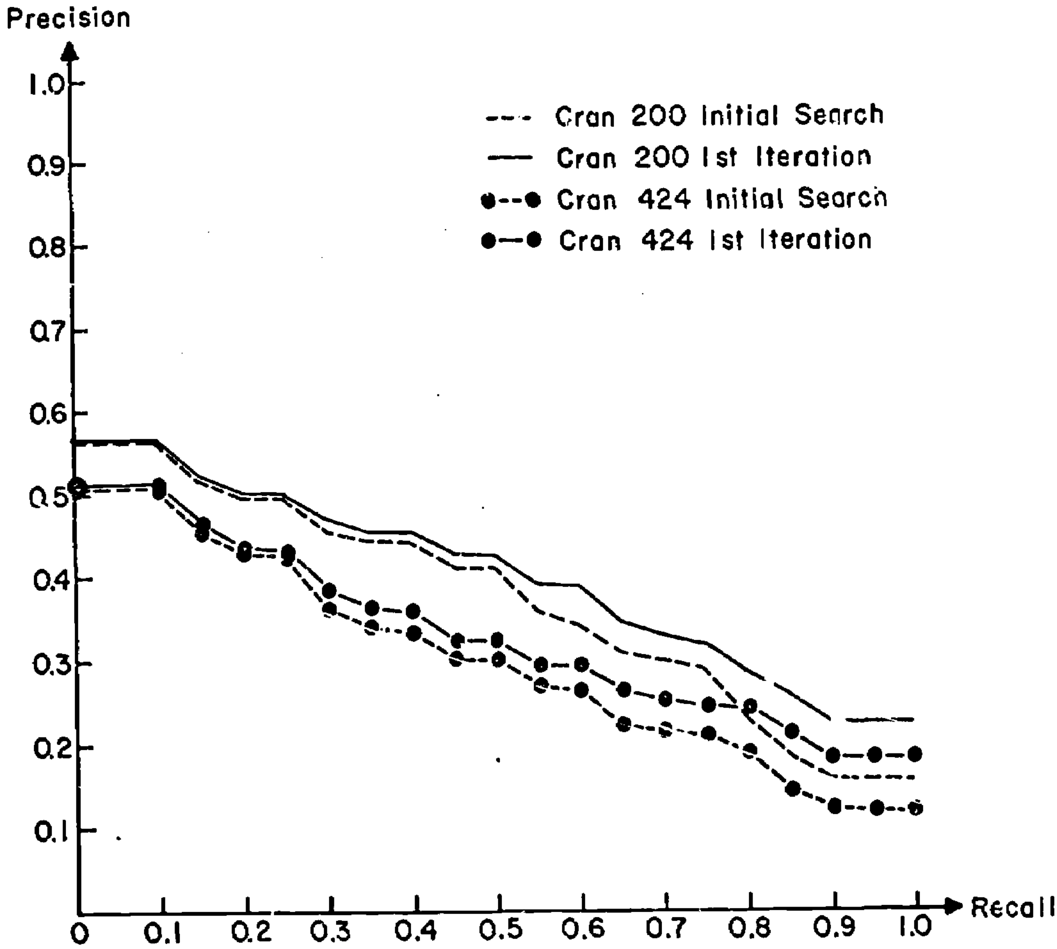
It may be noted that once again the recall-precision output favors the small collection, whereas the recall-fallout output is more favorable to the larger collection. Furthermore, while the generality decreases by a factor of over 2 from small to large collection, the fallout drops by less than one-half. These results are entirely in agreement with those previously obtained for the Cranfield 1400 collection. The output of Figs. 3 and 4 for the positive feedback strategy also indicates that the magnitude of improvement provided by one feedback iteration is approximately comparable for the two collections.

In order to investigate the question of feedback improvement in more detail, several feedback procedures were tested including, in particular, the following three types (based on the retrieval of the top five documents in each case):

Property	Cranfield 200	Cranfield 424
Source	Abstracts in aerodynamics	Abstracts in aerodynamics
Analysis	Word stem process	Word stem process
No. Documents	200	424
No. Queries	22	22
No. of Relevant	115	115
Search	Feedback search	Feedback search
Generality	.0261	.0123
Ave. Fallout	.0333	.0211

Collection Properties for Feedback Searches
Using Cranfield 200 and 424

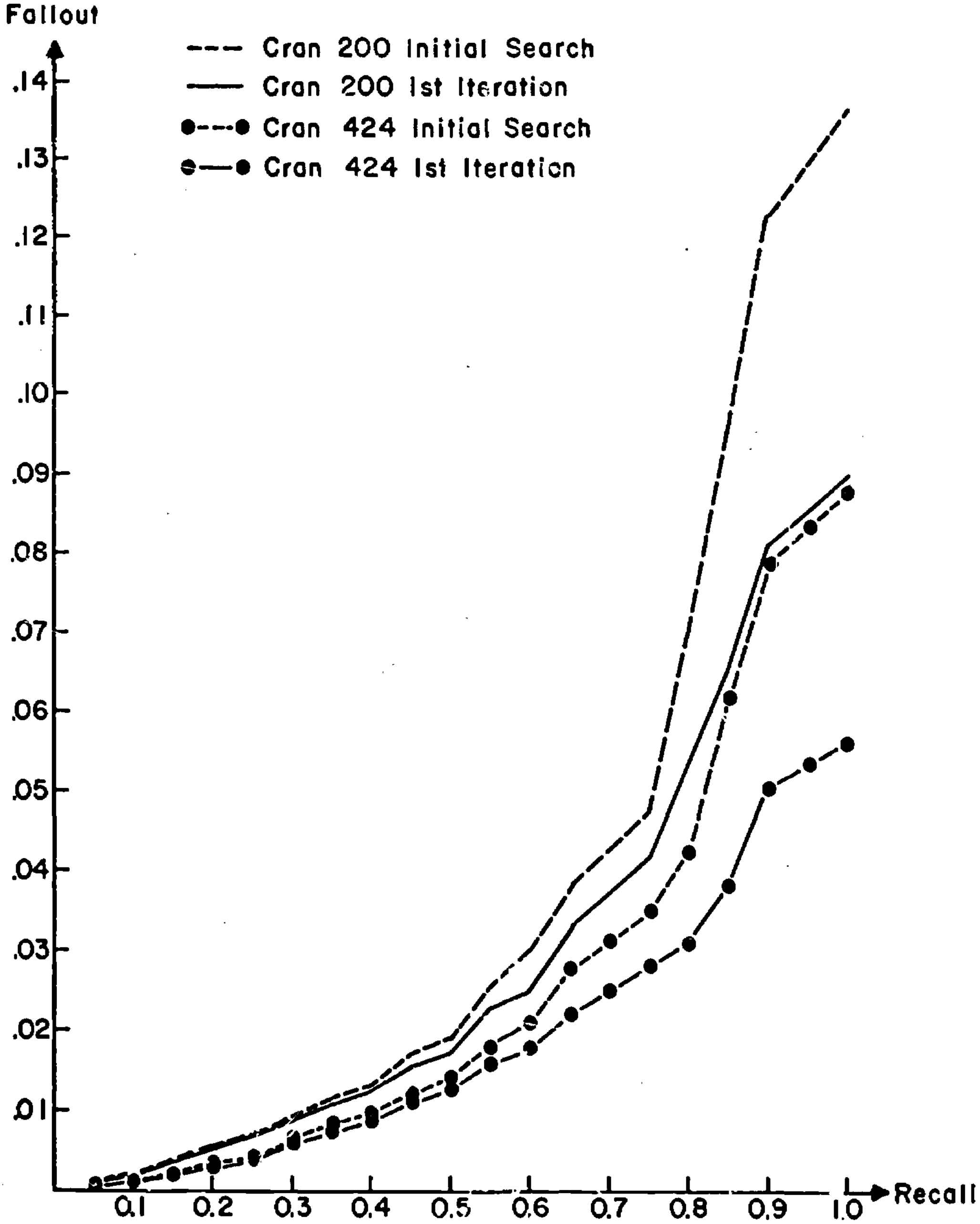
Table 6



Recall-Precision Comparison for Cran 200 and 424 Collections

(initial run and one feedback iteration—positive feedback only, word stem process, 22 queries)

Fig. 3



Recall-Fallout Graph for Cran 200 ar ' 424
(Initial run and one feedback iterati...-
positive feedback only)

Fig. 4

a) positive feedback, where information obtained from documents known to be relevant is used to update the query formulation;

b) selective negative feedback, where positive information is derived from the relevant documents together with negative information obtained from the top retrieved nonrelevant item;

c) modified selective negative feedback, where the negative information derived from the nonrelevant documents is used only when no positive information is available.

The evaluation is based principally on two evaluation functions, which measure respectively the precision improvement and the fallout improvement as follows: [12]

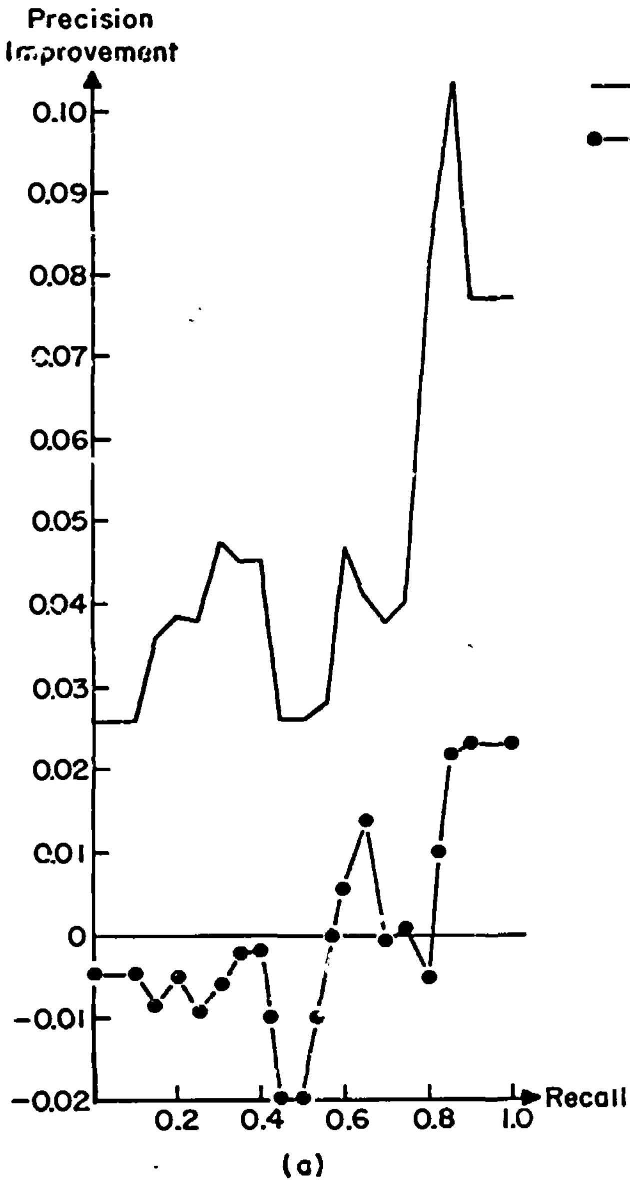
$$\text{Precision improvement} = P_1 - P_0 ,$$

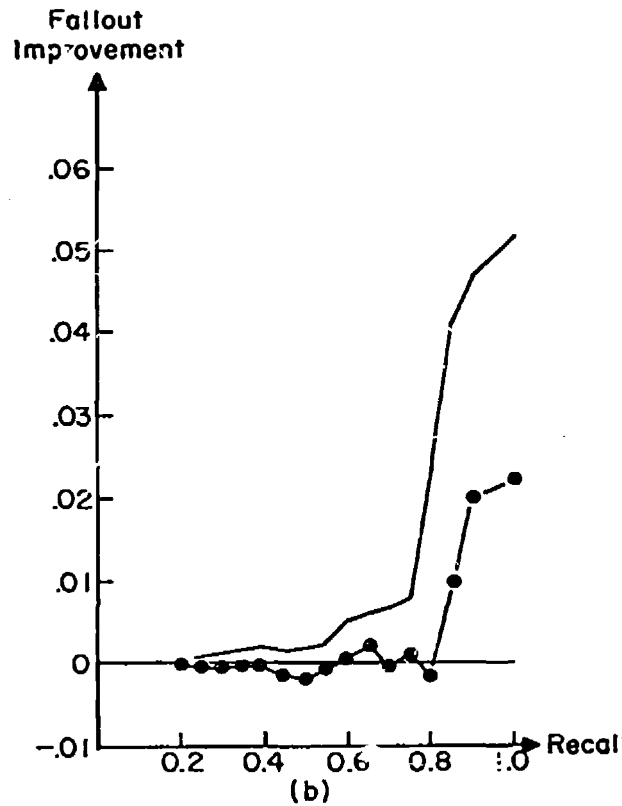
where P_0 is the precision of the initial search, and P_1 is the precision of the feedback iteration at a specified fixed recall point; and

$$\text{Fallout improvement} = F_0 - F_1 ,$$

where F_0 is initial fallout, and F_1 the fallout of the feedback iteration. (A performance improvement implies that the fallout for the feedback iteration is smaller than the initial fallout.)

The output for a selective negative feedback strategy which does not operate satisfactorily in an environment of decreasing generality is shown in Fig. 5. It is seen that for the larger collection the precision improvement is negative for most recall points, showing that the feedback process in fact hurts the performance. The same is true for some points of the fallout improvement curve. Apparently, the strategy represented by





Precision (a) and Fallout (b) Improvement for Selective Negative Strategy 5

(averages over 22 queries)

Fig. 5

the curves of Fig. 5 uses too many nonrelevant items for feedback purposes thereby hurting retrieval. (Fewer relevant items are retrieved early in the search for the Cran 424 collection, than for Cran 200.)

The performance for two feedback strategies which operate excellently with decreases in generality is shown by the precision and fallout improvement curves of Figs. 6 and 7. Fig. 6 covers the positive-feedback strategy which is seen to operate equally well for both collections. Still larger improvements are noted in Fig. 7 for the modified negative strategy in which a nonrelevant item is used for feedback purposes only when positive information (in the form of relevant retrieved documents) is not available.

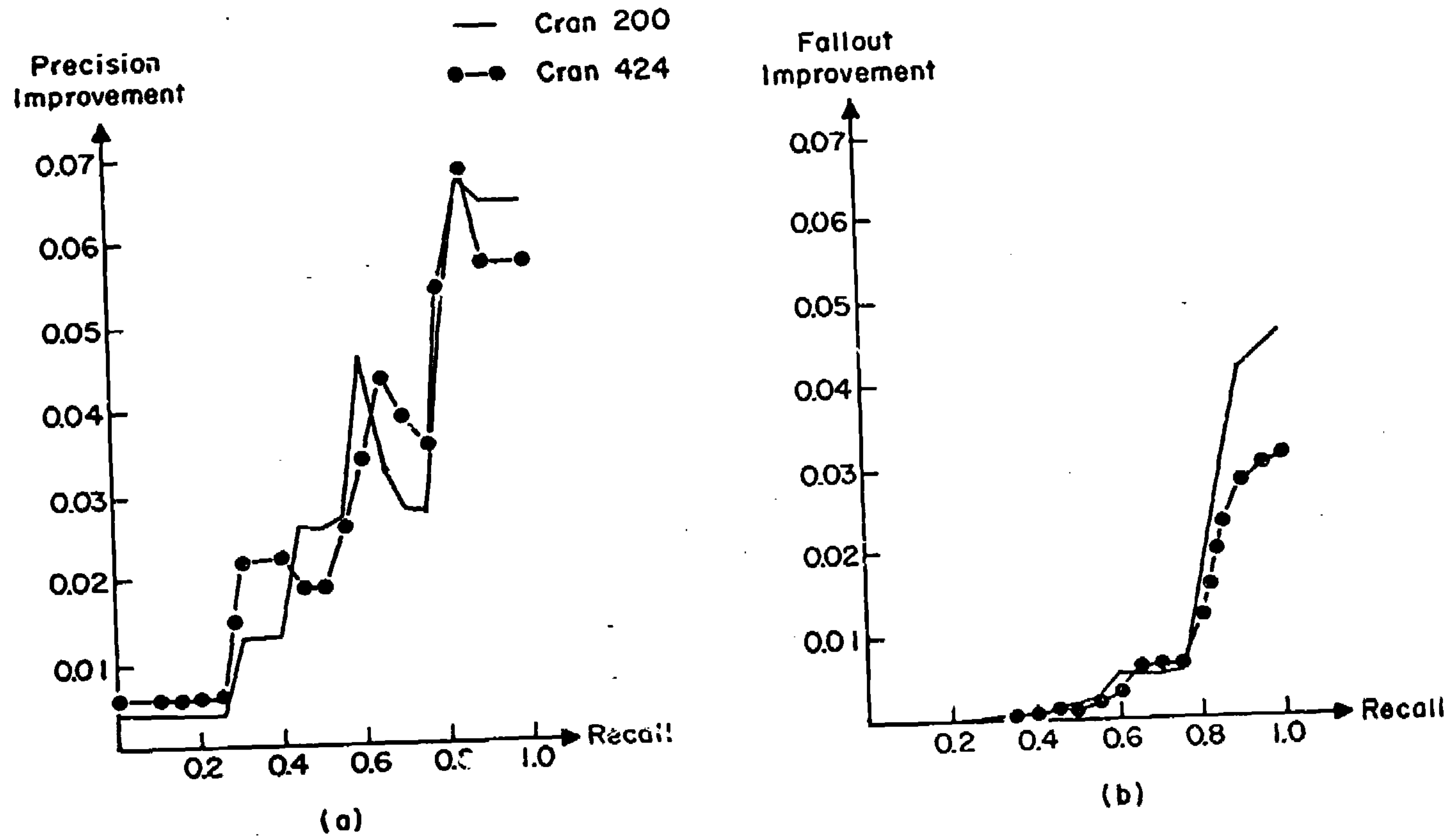
From the output of Figs. 6 and 7 it appears that feedback strategies can be implemented which operate equally well for collections of low and high generality. These strategies should be implementable in a realistic environment comprising thousands of items where they may be expected to produce the performance improvements previously noted for small test collections.

4. Variations in Relevance Judgments

A generality problem arises not only when collections of different size but identical relevance properties are to be compared, but also when the same collection is processed with different types of relevance assessments. In a previous study, a collection of 1268 documents in library science and documentation was examined using four types of relevance grades:

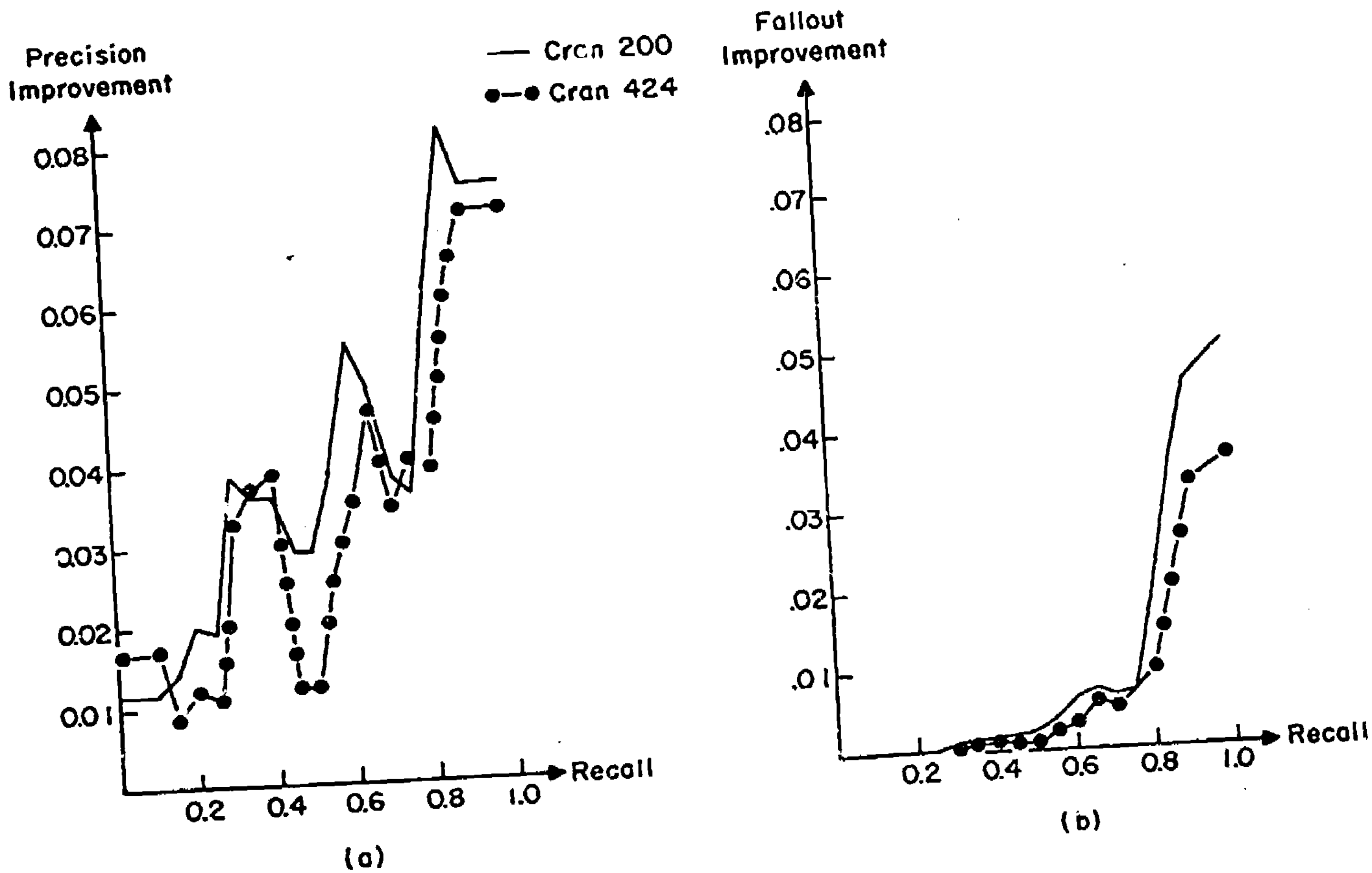
a) the A judgments representing relevance assessments by the query authors;

b) the B judgments representing nonauthor judges;



Precision (a) and Fallout (b) Improvement for Feedback Strategy I (positive feedback)
(averages over 22 queries)

Fig. 6



Precision (a) and Fallout (b) Improvement for Feedback Strategy 4
(modified selective negative strategy; averages over 22 queries)

Fig. 7

c) the C judgments representing the disjunction between the A and B judgments (that is, a document is judged relevant to a query if <u>either</u> A <u>or</u> B judges termed it relevant);

d) the D judgments representing the conjunction between A and B judgments (a document is judged relevant if <u>both</u> A <u>and</u> B judges termed it relevant).

It was demonstrated in the previous study [13], that the recall-precision performance graphs are relatively invariant to the variations caused by the multiple relevance assessments, and by the resulting changes in generality.

In an attempt to determine whether the performance characteristics obtained with collections of different size can be related to those produced by collections with varying relevance properties, the C and D collections are processed once again under slightly modified conditions. The collection properties are outlined in Table 7.

It will be noted that in the present case the generality change is produced not by adding any documents to the C collection in order to obtain the other collection of lower generality, but rather by subtracting from the set of relevant documents a number of items about which a unanimity of opinion could not be obtained by the relevance assessors. Nevertheless, the performance figures given in Table 7, and in Fig. 8(a) show that once again somewhat better recall-precision data for the collection of high generality (the C collection) are coupled with somewhat better fallout data for the collection of low generality (the D collection).*
This reflects the fact, on the one hand, that precision varies somewhat

*The recall-precision figures shown in Fig. 8(a) are not directly comparable to those produced in the earlier study [13] because of a small difference in the method used to produce performance averages over the total number of queries.

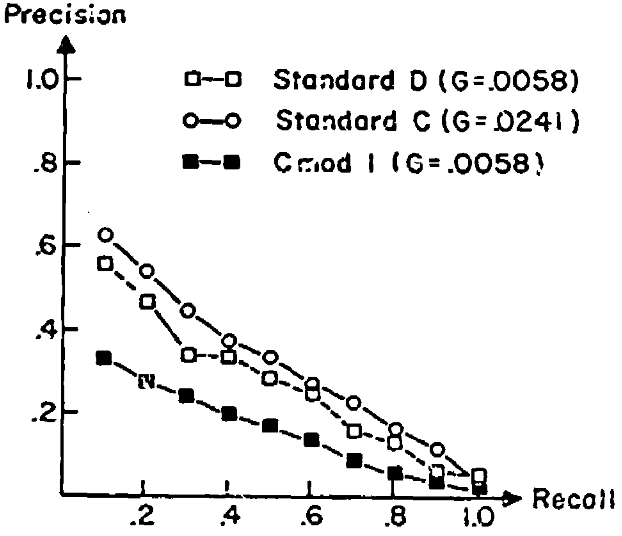
Property	Ispra C	Ispra D
Source	Document abstracts in documentation	Document abstracts in documentation
Analysis	Thesaurus	Thesaurus
No. Documents	1268	1268
No. Queries	45	45
No. of Relevant	1260	306
Search	Full search	Full search
Generality	.0241	.0058
Average Fallout	.1409	.0819

Collection Properties for Ispra C and D Collections

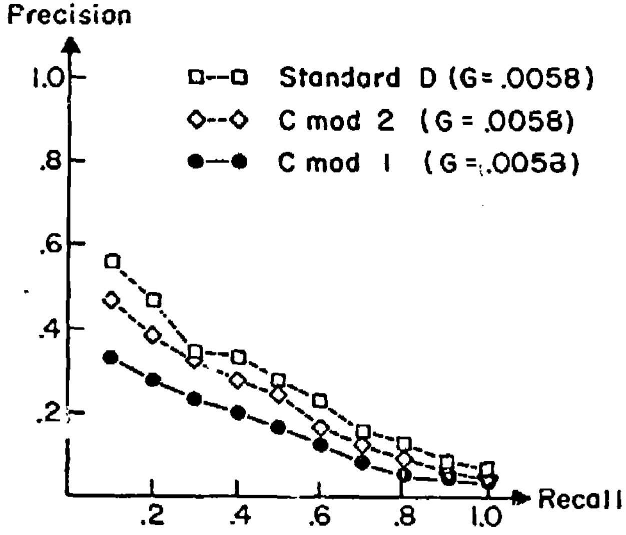
Table 7

R	P: D	P: C	P: C mod 1
0.1	.579	.627	.343
0.3	.344	.447	.234
0.5	.291	.334	.177
0.7	.155	.213	.099
0.9	.079	.111	.055

R	P: D	P: C mod 2	P: C mod 1
0.1	.579	.481	.343
0.3	.344	.332	.234
0.5	.291	.251	.177
0.7	.155	.127	.099
0.9	.079	.057	.055



a) Comparison of Standard C and D



b) Comparison of Modified C Runs

Generality Comparison for Collections of Fixed size

C mod 1 : $n_1 - k$ relevant items randomly set nonrelevant

C mod 2 : n_2 relevant retained; remainder scattered

Fig. 8

with generality, and therefore the collection with higher generality is likely to produce better precision. On the other hand, the collection of low generality exhibits better relevance judgments, since at least two judges had to agree on the relevance of each document; there exists therefore a greater certainty about the relevance (or nonrelevance) of each document with respect to each query, which implies that the nonrelevant are easier to reject using the D relevance judgments.

In order to see how the performance data change under a generality transformation, the C collection with high generality (.0241) is reduced to the generality of the D collection (.0058) in two different ways:

a) collection C mod 1 is produced by taking 962 relevant documents chosen at random and calling them nonrelevant; this reduces the original set of 1260 relevant documents in C to a total of 306 relevant (equal to the number of relevant in D);

b) collection C mod 2 is produced by retaining 306 out of the 1260 originally relevant items; the remaining 962 formerly relevant items are assigned random ranks in the collection instead of being retained with the rank they initially possessed as in C mod 1).*

The performance of the modified C collections which now exhibit the same generality as the standard D is presented in the recall-precision graphs of Fig. 8(b). It is seen that when the generality is kept invariant, as it is for the three collections of Fig. 8(b), the collection with the most reliable relevance judgments (the standard D) produces the best performance. Of the two modified C collections obtained by the generality

*The reranking process followed is described in a note by Williamson. [14]

transformation, the second produces better output than the first, since it is more carefully constructed by randomly deleting relevant items, and then randomly reintroducing them as nonrelevant ones with new ranks.

5. Summary

A variety of retrieval tests were performed with collections of varying generality in the areas of aerodynamics and documentation. Since precision varies with generality, the precision output generally favors the (small) collection of high generality. However, as the generality drops by a factor of k, the precision drops by a much smaller factor, and the fallout, which had been thought to remain invariant with generality changes, in fact decreases with generality, and thus favors the (large) low generality collections.

No clear extrapolation appears possible at this time which would permit a prediction to be made about the likely performance of very large collections of several hundred thousand items. However, the fallout data obtained in this study make it clear, that an argumentation which claims that the retrieval of 20 nonrelevant items for a collection of 1000 items would necessarily lead to an expected retrieval of 20,000 nonrelevant for a collection of a million is fallacious, since it assumes a constant fallout performance.

The user feedback procedures appear to be useful for collections of varying generality, and they should be implemented in operational environments. Finally, when generality variations arise from inconsistencies in the relevance assessments, the collection with the most secure relevance data performs best.

As larger document collections come into experimental use, the

fallout and precision figures should continue to be compared with the generality variations. In this fashion, it may be possible, in time, to obtain reliable projections for the performance with large collections under operational conditions.

References

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

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

[3] G. Salton and M. E. Lesk, Computer Evaluation of Indexing and Text Processing, Journal of the ACM, Vol. 15, No. 1, January 1968.

[4] G. Salton, Automatic Processing of Foreign Language Documents, Journal of the ASIS, Vol. 21, No. 3, May-June 1970.

[5] G. Salton, A Comparison between Manual and Automatic Indexing Methods, American Documentation, Vol. 20, No. 1, January 1969.

[6] E. M. Keen, Evaluation Parameters, Report No. ISR-13 to the National Science Foundation, Section II, Department of Computer Science, Cornell University, January 1968.

[7] S. E. Robertson, The Parametric Description of Retrieval Tests – Part 1: Basic Parameters, Journal of Documentation, Vol. 25, No. 1, March 1969.

[8] D. R. Swanson, Implications of the SMART-Medlars Comparison for Very Large Collections, unpublished manuscript, University of Chicago, 1970.

[9] R. G. Crawford, A Generality Study, Project Report, Computer Science Department, Cornell University, January 1970.

[10] G. Salton, Search and Retrieval Methods in Real-Time Information Retrieval, Proc. IFIP Congress 68, North Holland Publishing Company, Amsterdam 1969, p. 1082-1093.

[11] M. E. Lesk and G. Salton, Interactive Search and Retrieval Methods Using Automatic Information Displays, Proc. AFIPS Spring Joint Computer Conference, AFIPS Press, Montvale, N. J., 1969, p. 435-446.

[12] B. Capps and M. Yin, Effectiveness of Feedback Strategies on Collections of Differing Generality, Scientific Report ISR-18, Department of Computer Science, Cornell University, October 1970.

[13] M. E. Lesk and G. Salton, Relevance Assessments and Retrieval System Evaluation, Information Storage and Retrieval, Vol. 4, No. 4, 1969, p. 343-359.

[14] R. Williamson, A Proposal for an Experiment to Ascertain the Relationship between the Generality Ratio and Performance Measures, unpublished notes, Cornell University, 1969.

III. Automatic Indexing Using Bibliographic Citations

G. Salton

Abstract

Bibliographic citations attached to technical documents have been used variously to refer to related items in the literature, to confer importance to a given piece of writing, and to serve as supplementary indications of document content. In the present study, citations are used directly to identify document content, and an attempt is made to evaluate their effectiveness in a retrieval environment. It is shown that the use of bibliographic citations in addition to the normal keyword-type indicators produces improved retrieval performance, and that in some circumstances, citations are more effective for retrieval purposes than other more conventional terms and concepts.

1. Significance of Bibliographic Citations

The role of bibliographic citations attached to scientific and technical documents has received intensive study for many years. Several authors have noted, in particular, that the number of incoming citations (that is, the number of citations from a given set of outside documents to a specified target document) constitute useful indicators of document type and importance [1,2]. In consequence, the so-called "bibliographic network" consisting of documents and citations between them has been used to assess the characteristics of scientific and technical communications. [3]

In addition to providing indications of document influence,

bibliographic citations also play a role as content identifiers. The close affinity between the citations attached to a given document and the normal keyword-type content indicators has been expressed by Garfield in the following terms [4]:

> "By using the author's references in compiling the citation index, we are in reality using an army of indexers, for every time an author makes a reference, he is in effect indexing that document from his point of view...."

Furthermore, only a very small proportion of documents appears to be totally disconnected from the bibliographic network, in the sense that these documents do not cite any other documents nor are they cited from the outside [3]:

> "...there is a lower bound of one percent of all papers that are totally disconnected in a pure citation network, and could be found only by topic indexing...."

As a result, search tools such as the "citation index" which lists all incoming citations for each document in the index have proved to be useful adjuncts to information search and retrieval.

A variety of studies have been undertaken in an attempt to determine the relationship between standard keywords and bibliographic citations for content analysis purposes. Thus, it was determined that papers which were related by similarities in bibliographic citation patterns also provided a large number of common subject identifiers. [5] Furthermore, the correlation between citation similarities on the one hand, and index term similarities on the other is found to be far greater than expected for random document sets. [6]

While bibliographic citations appear not to have been used directly

as content indicators for retrieval purposes up to the present time, a number of experiments have been performed in which citations were incorporated as feedback information during the search process, in an attempt at retrieving additional information similar to that being identified in the search. [7,8] Specifically, an initial search would be made, leading to the retrieval of a number of documents. These would be scanned by the user, and information about these documents — including in particular document authors, citations made by the documents, and authors of these citations — would be returned to the system to be incorporated into an improved search formulation. The evaluation of this bibliographic feedback process proved, in particular, that [8]:

> "...no differences greater than four percent were found between the results of feeding back only subject data, and those of feeding back only bibliographic data. This implies that the usefulness of bibliographic data for feedback is of the same order as that of subject descriptors."

In addition, the same study showed that when citation data were added to standard subject indicators in a feedback environment, improvements of up to ten percent in retrieval effectiveness were obtained over and above the results produced by subject information alone. This led to the conjecture that [8]:

> "Since the bibliographic information is useful for feedback purposes, it should also prove valuable for initial retrieval searches."

An attempt is made in the remainder of this study to evaluate the correctness of this statement. Specifically, a collection of 200 documents in the field of aerodynamics is processed against a set of 42 queries using

first the normal content analysis methods incorporated into the automatic SMART document retrieval system [9], and then a modified process based on the bibliographic citations attached to the documents. The test design and evaluation results are covered in the remaining sections of this report.

2. The Citation Test

Consider a given document collection available in the form of English language abstracts, together with a corresponding set of user queries. Given such a collection, various linguistic analysis procedures may serve to reduce each item into analyzed vector form. A concept vector, representing either a document or a query, normally consists of a set of terms, or concepts, together with the respective concept weights. Two of the content analysis methods most frequently used with the SMART retrieval system are the word stem, and the thesaurus processes. In a word stem analysis, each concept incorporated into a normal concept vector represents a word stem extracted from the document, whereas for the thesaurus procedure, the concepts represent thesaurus categories obtained by consulting an automatic dictionary during the analysis operations. Word stems, or thesaurus categories are then concepts somewhat similar to the standard subject indicators normally assigned manually to queries and documents. In such an environment, the normal retrieval operation would consist in matching the concept vectors for queries and documents, and in retrieving for the users' attention all documents whose vectors exhibit a reasonable degree of similarity with the corresponding query vectors.

If it is assumed that each document carries with it a set of bibliographic citations (either to or from the document), it is possible to add to the normal document concept vectors, suitably chosen codes representing the bibliographic citations; alternatively, the citation codes might replace the normal concepts.

In order to obtain a match between citation codes attached to documents and normal user queries, it becomes necessary to attach citation information also to the queries. This can be done in one of two ways:

a) some queries may have been formulated by the user population in response to a set of documents known in advance to be relevant; that is, for each query one or more source documents exist, and the user's query is designed to retrieve additional items similar to the respective source documents*;

b) alternatively, a source document does not exist in advance, but the user is able to designate some other document as likely to be relevant to his query.

In either case, it becomes possible to add to the query vectors citation codes corresponding to source document citations, or to citations attached to the designated relevant documents, as the case may be.

These operations then produce expanded query and document vectors consisting partly of standard concept codes, and partly of citation codes, as shown schematically in Figure 1. Three types of retrieval operations become possible:

a) using only standard subject identifiers (the 'x' concepts of Figure 1);

b) using only citation concepts (the 'y' concepts of Figure 1);

c) using both the standard and the citation concepts (the 'x'

*In a previous test in which original query formulations were replaced by source document vectors, it was shown that the retrieval effectiveness produced by the source document "queries" was substantially better than that obtained with the standard queries. [10]

x x x x x x x x x x x x x x	y y y y y y
normal document concepts	bibliographic citat

a) Typical Expanded Document V

x x x x x x	y y y y y y
normal query concepts	citations to source document

b) Typical Expanded Query Vector

Expanded Query and Document Vectors

Figure 1

and 'y' information).

In these circumstances, the relative value of the citation information may be ascertained by comparing the results obtained with these three types of concept vectors.

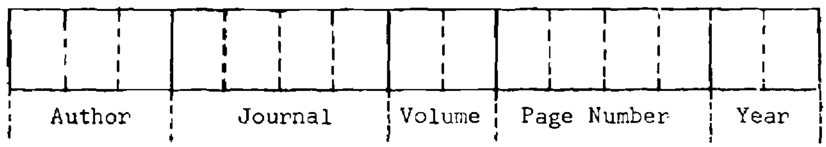
For the test under discussion, a collection of 200 document abstracts in aerodynamics was used with 42 search requests obtained from research workers in aerodynamics (the Cranfield collection [11]). Each document carried an average of 18 bibliographic references (outgoing citations to other documents), and each query was originally formulated in response to a source document. The set of source documents were similar in nature to the standard documents, in the sense that bibliographic citations were available for each; however, no source document was included among the standard 200.

To generate the citation portion of the document and query vectors, each citation was represented by a 15-character code. The citation coding is outlined in Figure 2, and some encoded sample documents are exhibited in the appendix. In order to increase the similarity coefficient for all documents cited by the query source documents, a citation code was added to each document vector not only for all outgoing citations, but also for each of the original documents. That is, each document is assumed implicitly to cite also itself (self-citation). A match between a query citation concept and a document citation concept may then be due to one of two causes:

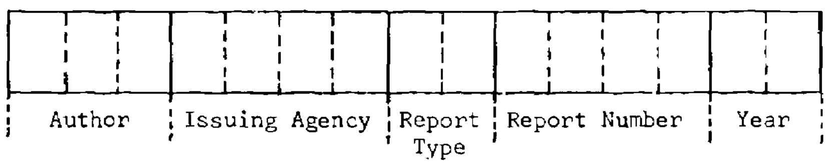
a) a request citation (source document citation) is identical with the document itself (request cites document);

b) a request citation is identical with a citation from a document (request and document have a common citation).

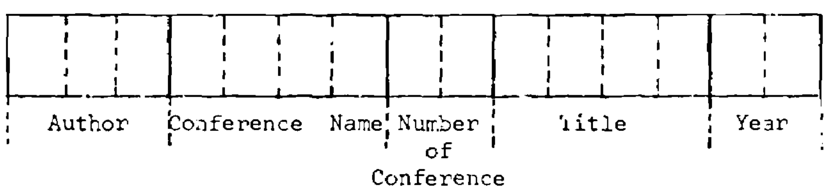
A comparison between citation effectiveness and standard concepts



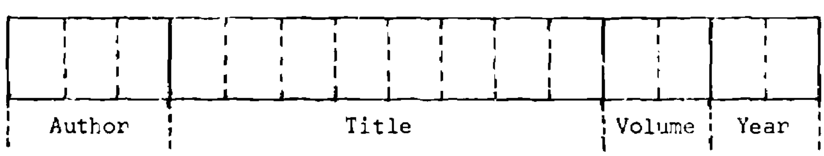
a) Typical Journal Code



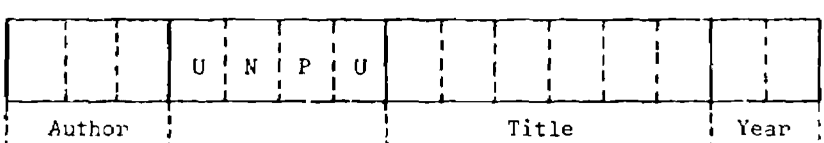
b) Typical Report Code



c) Typical Conference Paper Code



d) Typical Book Code



e) Unpublished Paper Code

Citation Coding

Figure 2

is obtained as usual by computing recall and precision values for the various runs while comparing the output.* The performance results are described in the remaining sections of this study.

3. Evaluation Results

The computation of recall and precision results depends on the availability of relevance assessments stating the relevance characteristics of each document with respect to each query. The original ("A") relevance assessments for the Cranfield collection were obtained by first submitting to the query authors for assessment the set of all documents cited by the source document, followed by additional items likely to be relevant. Since the source document citations were thus given special treatment, a bias may exist in favor of these citations – that is, an item cited by the source document may be more likely to be assessed as relevant than other extraneous documents. For this reason, three additional sets of relevance judgments were independently obtained from nonauthor subject experts, for which all documents were treated equally; that is, no special identification was provided for source document citations. The characteristics of the four sets of relevance assessments are summarized in Table 1.**

It may be seen that the four types of relevance assessments fall into two main categories as follows:

a) sets A and B have low generality characteristics – only four

---

*Recall is the proportion of relevant documents retrieved, and precision is the proportion of retrieved items actually relevant. Ideally one would like to retrieve all relevant and reject all nonrelevant to produce recall and precision values equal to 1. When recall is plotted against precision, as in a standard recall-precision graph, curves close to the upper right-hand corner represent superior performance, since both recall and precision are then maximized.

**The writer is indebted to Mr. C. W. Cleverdon for making available the Cranfield collection together with the various relevance assessments.

Relevance Judgments	Generality (Average Number of Relevant per Query)	Percent Overlap with "A" Judgments $\left[\frac{A \cap x}{A \cup x}\right]$
Original Judgments "A"	4.70	100.00%
"B" Judgments	4.28	80.74%
"C" Judgments	11.94	37.09%
"D" Judgments	11.70	37.83%

Relevance Assessments

Table 1

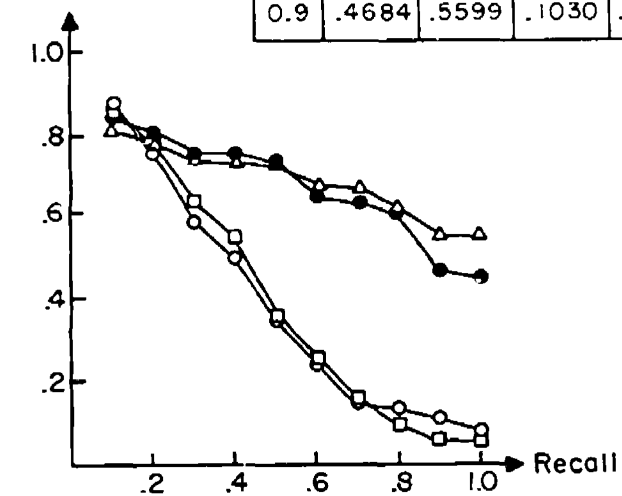
to five relevant items per query — corresponding to a strict interpretation of relevance; furthermore the A and B assessments are very similar in nature in view of the overlap of over 80 percent in the respective sets of relevant items per query;

b) sets C and D exhibit much higher generality — almost 12 relevant items per query — corresponding to a less narrow relevance interpretation, and the similarity with the original A judgments is much smaller.

Under normal circumstances, one would expect a better recall-precision performance for the high-generality case, while for equivalent generality, the best relevance assessments would produce the best performance [12]. The actual retrieval effect of the four types of relevance assessments is outlined in the graphs of Figure 3.

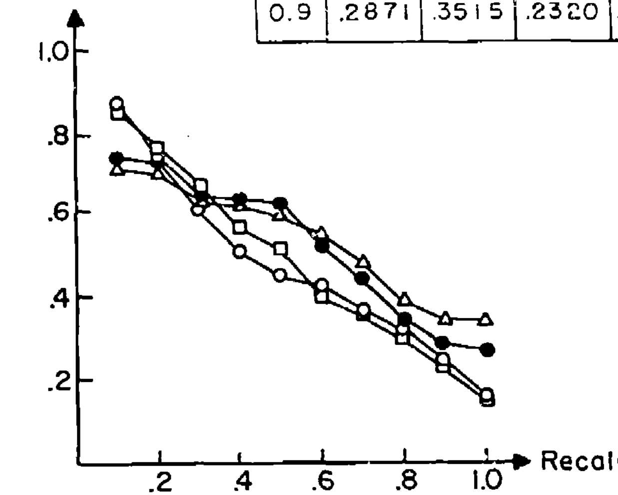
It may be seen that when citations only are used in query and document vectors (the 'y' portions), the low generality A and B assessments give much superior performance (Figure 3 (a)). On the other hand, when standard thesaurus concepts are used in addition to citations, as in Figure 3(b), the differences among the four types of assessments largely disappear. The same is true when the thesaurus alone is used for analysis purposes (without the additional citations). The latter results are in agreement with earlier studies showing that only minor differences occur in averaged recall-precision graphs with normal variations in relevance assessments. [13] The large differences in the performance of the "citations only" run of Figure 3(a) must then be due to the peculiar nature of relevance assessments 'A' and 'B', and to the special treatment accorded to the source document citations during the relevance judging procedure. For practical purposes, it appears safer to use the 'C' and 'D' judgments in assessing the relative importance of

R	Precision ●—●	Precision △—△	Precision ○—○	Precision □—□
0.1	.8402	.8135	.8833	.8683
0.3	.7611	.7496	.5899	.6350
0.5	.7474	.7261	.3292	.3392
0.7	.6139	.6628	.1277	.1562
0.9	.4684	.5599	.1030	.0586



a) Citations Only (Citations of Query Source Doc. Used as Query)

R	Precision ●—●	Precision △—△	Precision ○—○	Precision □—□
0.1	.7421	.7106	.8719	.8675
0.3	.6604	.6568	.6138	.6603
0.5	.6201	.6002	.4584	.5123
0.7	.4463	.4939	.3617	.3448
0.9	.2871	.3515	.2320	.2165



b) Thesaurus with Citations

Effect of Relevance Assessments on Citation Indexing
(200 documents, 42 queries)

●—● Original 'A' Relevance Judgments
△—△ 'B' Relevance Judgments
○—○ 'C' Relevance Judgments
□—□ 'D' Relevance Judgments

Figure 3

citation data and standard subject indicators in a retrieval environment.

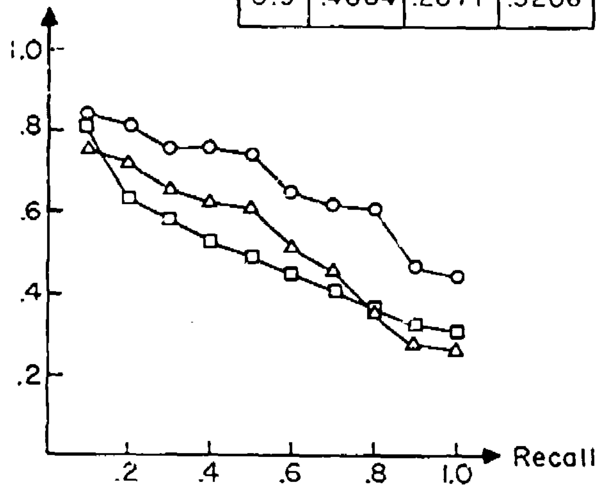
The main output results are shown in Figure 4 for both 'A' and 'C' relevance assessments. It may be seen that in both cases the augmented thesaurus vectors, obtained by adding citation concepts to standard subject indicators, improve the precision performance by up to ten percent for a given recall point. The short "citations only" vectors provide superior performance for the 'A' relevance assessments for the reasons already stated. Even with the 'C' judgments, the citation indexing alone provides a very high standard of performance in the low recall range.

The usefulness of bibliographic citations for content analysis purposes is further illustrated by the output of Figure 5 in which a standard word stem matching process is compared with the word stem vectors augmented by citation information. It can be seen from the output of Figure 5(a) that the augmented stem vectors generally produce better performance than the standard word stems; this confirms the results obtained in Figure 4 for the thesaurus process. Furthermore, the output of Figure 5(b) shows that augmented thesaurus vectors are slightly preferable to augmented word stem vectors.

The performance data of Figures 3 to 5 were obtained by adding source document citations to the normal query formulations. Since the source documents exhibit especially strong relevance characteristics — each user knows in advance that the source documents are immediately germane to the information queries — an attempt was made to relax the requirement for source document citations by replacing them by the citations attached to a randomly chosen relevant document.

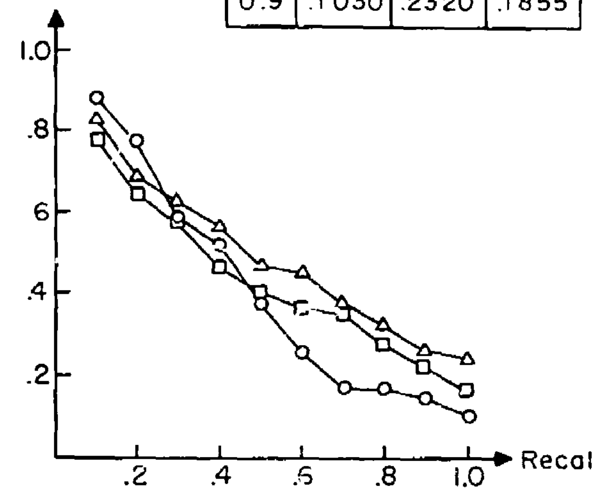
Specifically, each query is first processed in the standard manner using a normal thesaurus look-up procedure. A document identified as relevant

R	Precision		
	○—○	△—△	□—□
0.1	.8402	.7421	.8144
0.3	.7611	.6604	.5733
0.5	.7474	.6201	.4811
0.7	.6169	4463	.3849
0.9	.4684	.2871	.3206



a) 'A' Relevance Judgments

R	Precision		
	○—○	△—△	□—□
0.1	.8833	.8719	.7648
0.3	.5899	.6138	.5197
0.5	.3292	.4584	.4064
0.7	.1277	.3617	.3255
0.9	.1030	.2320	.1855

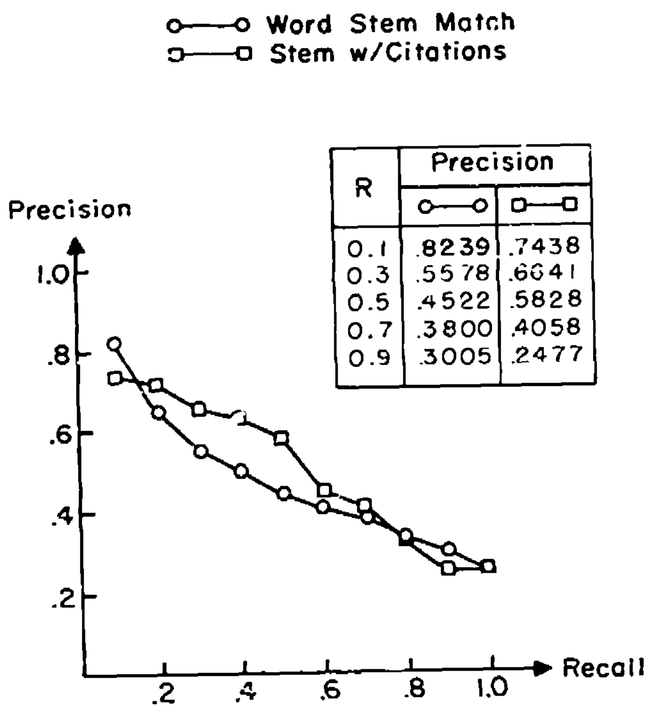


b) 'C' Relevance Judgments

Comparison of Citation Indexing with Thesaurus Operations
(200 documents, 42 queries; source documents)

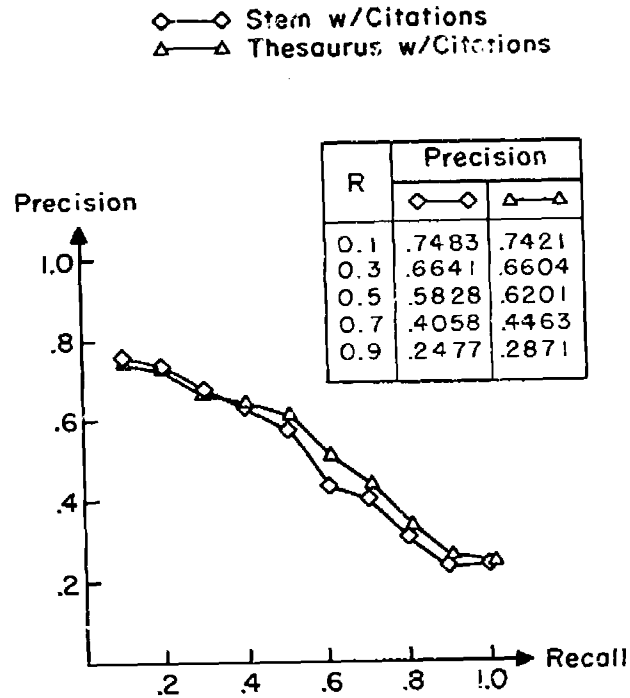
○—○ citations only
△—△ thesaurus with citations
□—□ thesaurus (Harris 3)

Figure 4



R	Precision: Word Stem Match (o—o)	Precision: Stem w/Citations (□—□)
0.1	.8239	.7438
0.3	.5578	.6641
0.5	.4522	.5828
0.7	.3800	.4058
0.9	.3005	.2477

a) Word Stems with Citations



R	Precision: Stem w/Citations (◇—◇)	Precision: Thesaurus w/Citations (△—△)
0.1	.7483	.7421
0.3	.6641	.6604
0.5	.5828	.6201
0.7	.4058	.4463
0.9	.2477	.2871

b) Word Stems vs. Thesaurus

Effect of Citations on Word Stem Matching Process
(200 documents, 42 queries; source documents)

Figure 5

after the fact — but not known to the user in advance — is then used in lieu of the normal source document, and citations from this relevant document are used to form the augmented query vector. The relevant documents chosen for this purpose are eliminated from the document collection for evaluation purposes. The output of Figure 6 shows that the citations obtained from the randomly chosen relevant documents do not have sufficiently strong relevance characteristics to lead to an improved retrieval performance over and above the standard thesaurus method.

The following principal results emerge from the present citation test:

a) the general usefulness of bibliographic citations for document content analysis, previously noted by a number of other investigators, is entirely confirmed;

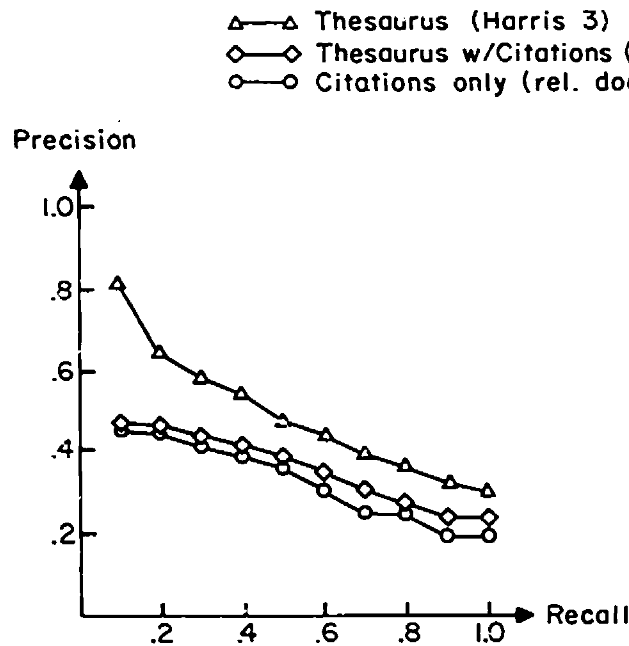
b) bibliographic citations used for document content identification provide a retrieval effectiveness fully comparable to that obtainable by standard subject indicators at the low recall-high precision end of the performance range;

c) the augmented document vectors, consisting of standard concepts plus bibliographic citation identifiers appear to provide a considerably better retrieval performance than the standard vectors made up of normal subject indicators only;

d) the bibliographic citations attached to information requests should be taken from documents whose strong relevance characteristics to the respective queries is known in advance by the user population.

The present experiment then leads to the conclusion that documents processed in a retrieval system should normally carry bibliographic citation codes in addition to standard content indicators. When queries are received from the user population, improved service can be obtained by using

ument citations as part of the query formulations whenever documents with



R	Precision		
	△—△	◇—◇	○—○
0.1	.8144	.4680	.4390
0.3	.5733	.4364	.4112
0.5	.4811	.3915	.3615
0.7	.3849	.3084	.2518
0.9	.3206	.2361	.1935

Use of Citations from Random Relevant Documents
(200 documents, 42 queries)

Figure 6

a priori relevance characteristics are identified by the users at the time of query submission. If no documents with strong relevance characteristics are available when the query is first received, bibliographic citations can still be used as a feedback device by updating the query formulations with citations from previously retrieved relevant documents.

References

[1] J. H. Westbrook, Identifying Significant Research, Science, Vol. 132, No. 3435, October 28, 1960, p. 1229-1234.

[2] J. Margolis, Citation Indexing and Evaluation of Scientific Papers, Science, Vol. 155, No. 3767, March 10, 1967, p. 1213-1219.

[3] D. J. de Solla Price, Networks of Scientific Papers, Science, Vol. 149, No. 3683, July 30, 1965, p. 510-515.

[4] E. Garfield, Citation Indexes for Science, Science, Vol. 122, No. 3159, July 15, 1955, p. 108-111.

[5] M. M. Kessler, Comparison of the Results of Bibliographic Coupling and Analytic Subject Indexing, American Documentation, Vol. 16, No. 3, July 1965, p. 223-233.

[6] G. Salton, Associative Document Retrieval Techniques using Bibliographic Information, Journal of the ACM, Vol. 10, No. 4, October 1963, p. 440-457.

[7] J. W. McNeill and C. S. Wetherell, Bibliographic Data as an Aid to Document Retrieval, Scientific Report No. ISR-16 to the National Science Foundation, Section VIII, Cornell University, September 1969.

[8] M. Amreich, G. Grissom, D. Michelson, E. Ide, An Experiment in the Use of Bibliographic Data as a Source of Relevance Feedback in Information Retrieval, Report No. ISR-12 to the National Science Foundation, Section XI, Cornell University, June 1967.

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

[10] R. G. Crawford and H. Z. Melzer, The Use of Relevant Documents instead of Queries in Relevance Feedback, Scientific Report No. ISR-14 to the National Science Foundation, Section XIII, Cornell University, October 1968.

[11] C. W. Cleverdon and E. M. Keen, Factors Determining the Performance of Indexing Sustems, Vol. 2, Test Results, Aslib Cranfield Research Project, Cranfield, 1966.

[12] G. Salton, The Generality Effect and the Retrieval Evaluation for Large Collections, Scientific Report No. ISR-18 to the National Science Foundation and to the National Library of Medicine, Section II, Cornell University, October 1970.

[13] M. E. Lesk and G. Salton, Relevance Assessments and Retrieval System Evaluation, Information Storage and Retrieval, Vol. 4, No. 4, October 1968, p. 343-359.

Appendix

Sample Citation Codes

I. Sinnott, Colin S., "On the Prediction of Mixed Subsonic/Supersonic Pressure Distributions," Journal of Aerospace Sciences, Vol. 27, p. 767, 1960.

SINOJAS27076760

II. Herriot, John G., Blockage Corrections for 3-Dimensional Flow Closed-Throat Wind Tunnels, with Consideration of the Effect of Compressibility, NACA, Rep. 995, 1950.

HERNACAORO99550

III. Cheng, H. K., "Hypersonic Shock-Layer Theory of the Stagnation at Low Reynolds Number," Proceedings of the 1961 Heat Transfer and Fluid Mechanics Institute, Stanford University Press, Stanford, California, 1961.

CHEPHTFOOHYPE61

IV. Couper, J.E., The Operation and Maintenance of Recorder Type IT 3-16-61, Unpublished M.O.A. Report.

CØUUNPUØPERAT**

V. Goldstein, S., Ed., Modern Developments in Fluid Dynamics, Vol. I, p. 135; Oxford, The Clarendon Press, 1938.

GØLMØDERNDE0138

IV. Automatic Resolution of Ambiguities from Natural Language Text

S. F. Weiss

Abstract

This study investigates automatic disambiguation by template analysis. The evolutionary process by which ambiguities are created is discussed. This leads to a classification of ambiguities into three classes: true, contextual, and syntactic. The class assigned to a given word is dependent on the syntactic and semantic functions performed by the word. Only true ambiguities are suitable for automatic resolution.

In this study, automatic disambiguation is accomplished by an extended version of template analysis. The process consists in locating an ambiguous word and in testing its environment against a predetermined set of rules for occurrences of words and structures which indicate the intended interpretation. Experiments using this process show that a high degree of accuracy in resolution can be achieved.

The process under consideration is not completely automatic because it requires that a set of disambiguation rules be created *a priori*. The creation of this rule set, however, is sufficiently straight forward that it may eventually be done automatically. A learning program is implemented to accomplish this. The process reads input words and attempts to resolve any existing ambiguities. If a resolution of the ambiguity is performed incorrectly, the rule set is augmented and modified appropriately, and the next input is considered.

The experimental results obtained are poor for the first few inputs. The performance steadily improves as more inputs are processed, and finally

levels off at above 90% accuracy. A true learning process is thus indicated.

The proposed learning process is not only useful for disambiguation, but can also serve for a number of other applications, where it may be desired to tailor a process to a particular user need.

1. Introduction

An ambiguous word is defined as a word which can have two or more different meanings. There exist a great many such ambiguous words and their occurrence in text is fairly common. In general they create no problem for a human reader because he is constantly aware of the context of the material he is reading and of the real world. This usually makes obvious the proper definition of an ambiguous word. For example, the word BOARD may mean, among other things, a piece of wood or a group of people. In the first of the two sample sentences below, the ambiguity is resolved by the context of the sentence while in the second, resolution is achieved by the reader's knowledge of the real world. In other words the reader knows from his general knowledge that it is much more likely to cut a piece of wood than a group of people, even though it is technically possible to do both.

A: He is a member of the board of directors.

B: He cut up the board.

Disambiguation by computer is considerably more difficult. A computer does not automatically conceptualize the context of the text as it is read. Also a computer cannot be expected to contain the vast store of knowledge that a human reader possesses. This study presents some techniques for automatic semantic disambiguation of words from natural language text and the application of template analysis to this process. A complete discussion of template analysis

is presented in Weiss [16].

The justification for such a study is that ambiguities in text are detrimental to any natural language process which uses that text. The extent of the damage imposed by ambiguities varies with the natural language process as is shown by the three examples below.

1. In a SMART-like information retrieval system ambiguous words are assigned multiple concepts to represent their various possible definitions. Since only one of the definitions is is actually correct, this process adds erroneous material to the document and query vectors. But this is not a serious problem since ambiguous concepts are rare and thus make up only a small part of a document or query vector. Resolution of ambiguities makes a very small change in a concept vector and hence causes only a very small change in document-query correlations. Thus in a retrieval environment, ambiguities may not pose a very serious problem and are hardly worth resolving. Examples 2 and 3 present environments in which the consequences of ambiguities are more serious and disambiguation is more justified.

2. A serious problem in automatic syntactic analysis is that an analyzer may produce many analyses for a single input. It is very difficult if not impossible to determine the intended analysis from among this set. Thus syntactic analysis schemes which generate as few analyses as possible are clearly the most desirable. One cause of multiple analyses is words which have more than one syntactic role. For example, the word FLYING can be either a verb or an adjective. This in turn gives rise to several analyses of

   THEY ARE FLYING PLANES.

   Some systems perform semantic tests to determine which of the syntactic analyses is semantically feasible. An even better approach is to resolve ambiguities prior to syntactic analysis

thus reducing the number of analyses produced. It sometimes happens that syntactically ambiguous words are also semantically ambiguous. NEGATIVE for example is usually an adjective when it means NOT and a noun in the photographic context. Thus by resolving the semantic ambiguity, the syntactic ambiguity is also removed. In this way resolution of semantic ambiguity can reduce the number of analyses resultant from an automatic syntactic analysis scheme and hence simplify the task of determining the correct analysis.

3. In natural language command analysis or a natural language programming language, each statement must be mapped into a unique command or command sequence. Statements which due to ambiguities simultaneously specify more than one command sequence are unexecutable. Current programming languages such as FORTRAN and ALGOL deal with this problem simply by prohibiting all but the most trivially resolvable ambiguities (such as the minus sign which may be unary or binary). This is not possible in natural language command analysis and thus all ambiguities must be resolved before execution is possible.

These three examples show how the problems caused by ambiguities in natural language text vary according to the application. In the third example resolution is a necessity while it is more or less a convenience in the other two. In general it appears that at best, ambiguities do no harm and at worst they are disasterous. In no case do they ever seem to have constructive effects. Of course there are other examples of consequences of ambiguities but these three seem sufficient to justify further investigation into the area of automatic disambiguation.

2. The Nature of Ambiguities

Most words in isolation do not have a well defined meaning. The exact meaning of a word is formed by the interaction of the word and its context. Each word is both acted upon by its context and acts upon its context. The

action that a word performs on its context is called its semantic function. This can be thought of as a mathematical function with the word's context as its argument and the total meaning as its value. An example is presented in Figure 1 below.

Phrase: Bottom of the bottle
Word: Bottom
Semantic function: indicates lowest point in context
Context: "of the bottle"

Application of semantic function to context yields the value: lowest point in the bottle

Example of Semantic Function

Figure 1

Building on the concept of semantic function it is now possible to define three types of ambiguities. A word is a true ambiguity if it has two or more distinct semantic functions. An example is the word DEGREE. This may refer to a unit of temperature or angle as in "a 90 degree turn" or an award from a school as in "college degree". These are clearly two separate semantic functions. Some words have only one semantic function yet still appear ambiguous. This situation is produced when a single semantic function, acting on a variety of contexts, produces vastly different meanings. Such words are termed contextually ambiguous. As an example, the word CORE is considered ambiguous in the ADI dictionary. It refers to both a computer memory and the central part of something. However there is only one semantic function at work here and it designates central aspect of its context. A computer memory is at least

conceptually if not physically the center of a computer. Thus CORE is a contextual ambiguity according to the definition above.

A third type of ambiguity is syntactic ambiguity. The meaning of such a word is dependent upon its syntactic role. The meaning of ELABORATE, for example, differs somewhat depending on whether it is used as a verb or adjective. These differences in meaning, however, are generally just slight variations of a single semantic concept.

The classification of an ambiguous word into one of these categories is not a strictly defined process. The categories are not completely disjoint; and the ambiguous words themselves are in a constant state of evolutionary change much like biological evolution. A good example of the development of an ambiguous word can be seen in the word BOARD which can mean a piece of wood, a group of people (board of directors), or food (room and board). Originally board referred only to a piece of wood or a table. Because of their close relation to the table, the people who met there and the food served on it became associated with the board. In time this connection disappeared and BOARD currently appears to have three separate meanings. In general, ambiguities seem to stem from idioms and associations due to similarities such as between the food and the table on which it is served. These words gradually evolve into contextual and finally true ambiguities. Many of the words currently considered contextually ambiguous may eventually become true ambiguities. For example, it is conceivable that in the future, computer memories may no longer be considered a central element of the machine Thus CORE, shown previously to be a contextual ambiguity, may become a true ambiguity. As another example, consider the word LUNACY. It was originally thought that this form of insanity was caused by the moon and hence the name. however, the lunar influence is better understood, and there is no

connection between the disease and the moon. Thus the common stem LUNA represents an evolved ambiguity.

Before considering resolution of ambiguities, it is necessary to decide which type or types can and should be resolved. There are several criteria for this decision. First, does the resolution of the ambiguity add any additional information to that already known? Second, does the added information warrant the work involved to determine it? And finally, what harmful effects might be expected if the ambiguity were not resolved?

As shown above the meanings of the various forms of a syntactic ambiguity vary only slightly. Thus very little information is added if resolution is performed. Also, harmful effects caused by syntactic ambiguities are slight and occur only in special cases as is shown in the following example. Let A, B, and C be words with A syntactically ambiguous and having meanings in thesaurus classes 1 and 2 (see Figure 2). B and C are not ambiguous. B is in thesaurus category 1 and C is in 2. Leaving A unresolved, that is using only a single concept to represent A, would in effect combine categories 1 and 2. This would make B appear synonymous to C which is not really the case. However, as shown previously, the differences in meaning of the various forms of syntactic ambiguities are slight thereby necessitating categories 1 and 2 being very close in meaning. Thus combining B and C is not a particularly grave error. For this reason it appears unwarranted to resolve syntactic ambiguities.

WORDS	THES. CATEGORIES
A	1,2 (SYN AMB)
B	1
C	2

Sample Syntactic Ambiguity
Figure 2

As discussed previously, contextual ambiguities have only one semantic function. The differences in meaning are caused by the context rather than by the word itself. It is therefore questionable whether such words should be disambiguated at all. Also because contextual ambiguities derive much of their meaning from context, they may have a broad spectrum of meanings rather than the few discrete meanings possessed by most true ambiguities. Intuitively at least this seems to indicate that the resolution of contextual ambiguities is both more difficult and less precise than resolution of true ambiguities. Experiments in this area show this to be the case.

The remaining class, the true ambiguities, demonstrates the properties necessary to justify their resolution. The remainder of this study deals with techniques for automatic resolution of true ambiguities.

3. Approaches to Disambiguation

Many automatic natural language analysis systems have a facility for automatic disambiguation. For some this entails the use of semantic information to resolve syntactic ambiguities and hence reduce the number of syntactic parses. Other systems actually tackle the problem of true semantic ambiguities. This section discusses some of these approaches to automatic disambiguation.

The easiest solution to the problem is simply to ignore it. This approach is actually not as absurd as it initially appears. When the domain of discourse is sufficiently limited, many ambiguities disappear. This is the case with the information retrieval system implemented by Dimsdale and Lamson [3]. By limiting the subject area to the medical field, the problem of ambiguities solves itself. For example, the word CELL has a number of possible meanings (dry cell, jail cell, muscle cell). However, only one of these interpretations is appropriate to medicine; and thus in this context, CELL may be treated as an unambiguous word.

As mentioned previously, one possible application for automatic disambiguation is in indexing documents for information retrieval. There are a number of possible techniques. Some researchers, for example Ranganathan [10] and Mandersloot et. al. [4], suggest that ambiguous words be represented by a number of concepts which resolve the ambiguity. One of these additional concepts could be the hierarchical father of the word under consideration. For example, the ambiguity caused by the word TYPE could be resolved by adding the concept for PRINTING. SMART uses a different method. An ambiguous word is assigned the concepts of all its possible interpretations. The set of concepts then share the total weight. Thus SMART covers all possibilities and is guaranteed of having the correct concept. However it is also guaranteed of having some wrong concepts. This inclusion of error would appear to weaken the indexing scheme and hence damage retrieval; but this is not the case. The occurrence of ambiguous words is quite rare and hence the error introduced by the process represents only a very small part of a total concept vector. Thus the effect on results is very small. In addition problems can only be caused when a thesaurus is used that contains words which are synonymous to some but not all of the interpretations of an ambiguous word. Actual experiments reveal that the resolution of ambiguities in SMART concept vectors results in improvement of less than 1%. Thus the added effort required to resolve ambiguities in this type of information retrieval context seems unwarranted.

Some question-answering systems with a restricted data base are able to disambiguate simply by testing the various interpretations against the data base and choosing the one that is applicable. DEACON is an example of one such system [15]. A query such as the one below is ambiguous since Guam is an island and an aircraft carrier. But since DEACON's data

base deals with ships, the latter interpretation is chosen.

How many people are on Guam?

Other systems perform a similar type of disambiguation by using lists of true predicates. Coles' system, for example, tests the query against a set of truth values. Similarly the process used by Schank and Tesler tests various ambiguous interpretations for consistency with a set of real world attributes.

Another basic method for automatic disambiguation is to associate semantic features with each word in the lexicon. Rules, similar to syntactic rules, can then test various possible interpretations for semantic as well as syntactic wellformedness. One such system is Simmons' PROTOSYNTHEX [12]. Each word is associated with its semantic class. For example, "angry" is a type of emotion and "pitcher" is a type of person (baseball player) or a type of container. Ambiguities such as "pitcher" are resolved by testing its syntactic structure against a set of semantic event forms. These indicate possible valid relationships between semantic classes. The semantic event forms reveal, for example, that a person can have an emotion while a container cannot. Thus the disambiguation of "angry pitcher" is accomplished. Woods accomplishes disambiguation in much the same way. Syntactic and semantic features are attached to words; and rules indicate legitimate combinations of these features.

Lesk uses a similar approach in his proposed natural language analysis system, but with a unique statistical feature [7]. In his system words are assigned both syntactic and semantic role indicators by the dictionary. The parse then determines syntactic dependencies and tests them for semantic validity. Those interpretations which fail the semantic test are eliminated thus accomplishing some disambiguation. In addition, each interpretation

of each ambiguous word has associated with it the probability of the "correctness" of that interpretation. For example, in a sports text the word "base" would be much more likely to refer to a baseball base than to a military base; and probabilities may be assigned accordingly. During the syntactic analysis a number of possible parses are developed. The probability of correctness for each is the product of its constituent probabilities. In this way, interpretations with very low probabilities of being valid may be eliminated thus accomplishing another form of disambiguation.

The processes presented above use syntactic and semantic features to qualify the words and then employ a common rule list to govern word combination. A more detailed approach to disambiguation is to attach specific combination rules to each word. The need for this can be seen in the following simple example. Most noun phrases consisting of an adjective and a noun assume the basic features of the noun. The phrase may then be used anywhere that the noun is legal. For example, the phrase "folding money" may be used wherever "money" can be used. This is not true for "folding" which in some sense loses its identity when combined with the noun. Most of the systems which use a combination rule list can determine this property. There are, however, exceptions to this rule. Consider the phrase "Tompkins County". Here the word "Tompkins", acting as an adjective, dominates the phrase. It is all right to say "Buffalo is in a county" but "Buffalo is in Tompkins County" is semantically and geographically incorrect. Thus in this case the phrase assumes the properties of the adjective. To treat properly this and other similar cases, it is useful to associate combination rules with individual words rather than using a common rule list

for all words. Some of the automatic systems which employ this approach are those by Kellogg [6], and Quillian [9].

Kellogg's scheme assigns a set of data structures to each interpretation of each word. These include semantic features and selection restrictions. For a particular word the selection restrictions limit the words with which it can be associated to only those with specific semantic features. For example, the verb "talk" can take only an animate subject.

In Quillian's Teachable Language Comprehender, memory is represented as an interconnected network of nodes. The meaning of a phrase is determined by locating a path in the network from one constituent word to the other. For some phrases there are more than one legal path. This indicates an ambiguous phrase. Disambiguation is achieved by using the shortest path. This represents the most likely interpretation and is thus similar in approach to Lesk's statistical scheme.

The processes discussed so far deal with disambiguation as a tool in some sort of information retrieval or question-answering facility. Moyne [8] summarizes this type of disambiguation as falling into one of four interaction types: interaction with the lexicon, with the data base, with the general system capabilities, and if all else fails, interaction with the user. This last type is strictly a last resort measure but is very helpful when unresolvable ambiguities are encountered.

As shown above, much of the work in disambiguation deals with larger information retrieval and question-answering systems. But some work has also been done on ambiguities alone. In particular is the work by Stone [14], Coyaud [2], and Borillo and Virbel [1]. All these schemes are based on resolution of ambiguities by examination of semantic context. Associated with each word is a set of words and concepts which, if found near the ambiguous word, specify

a particular interpretation. Stone concentrates on the resolution of ambiguities in high frequency words such as "matter". The study by Borillo and Virbel represents the most detailed and complete discussion of disambiguation encountered in the literature. They discuss all forms of ambiguities, and present for each, the methods needed for resolution. Ambiguous words are divided into five classes:

1. key word
2. grammatical ambiguity
3. semantic ambiguity
4. combined semantic and grammatical
5. forced

The key words are words of variable importance whose resolution is not vital. The forced words are so important that all interpretations must be represented. The remainder are self explanatory. The third and fourth classes are most interesting and correspond roughly to the true ambiguities presented in the previous section. Resolution is achieved by examing some environment of the ambiguous word for certain structural or semantic clues. In addition, Borillo and Virbel give a suggested list of attributes for a disambiguation process. These are first, that the context of an ambiguous word should be scanned in closest to farthest order. Second, resolution rules should be weighted according to their probability of correctness. And third, the scope of the context should be variable from word to word.

Building on this introduction, the next sections present an automatic disambiguation scheme using the template analysis process. It is designed as a disambiguation package for a natural language conversational system and hence expected input is clearly restricted. In addition, each ambiguous word is treated separately and the relevant context of each word is quite limited. Thus the input seems applicable to template analysis.

4. Automatic Disambiguation

A) Application of Extended Template Analysis to Disambiguation.

Associated with each ambiguous word is a set of keywords or structures which identify the intended meaning. For example, if within the context of the word BOARD, there are references to "fir", "pine", or "oak", a wooden board is probably intended. If "chairman" or "meeting" occurs, board would be taken to mean a group of people. This key to the intended meaning of am ambiguous word is usually found in the immediate context of that word, often in the same sentence. The actual optimal scope of context varies from word to word. Borillo and Virbel indicate that in general, best results are obtained using large sentence groups (document abstracts). In some cases, however, this is too broad and permits erroneous resolution by matching the wrong key. For this reason the scope of context is defined here to be the sentence containing the ambiguous word. Each sentence containing an ambiguous word is scanned for a resolution key. This resolution key may be a word, group of words, or structure, which reveals the intended meaning. The process is implemented using an extended version of template analysis [16]. This section discusses the extensions to template analysis that are required to facilitate automatic disambiguation. The disambiguation process is presented in subsection B and the experimental results in subsection C.

A template is basically a string of words. It matches a natural language input only if a substring of the input matches the template elements exactly including ordering and contiguity. Many ambiguities may be resolved using templates; but for others, templates are too strict a criterion. For these words the presence of a resolution key anywhere in the input is sufficient to warrant resolution. For this reason the context rule is used. Like a template, the context rule is a string of words. However a context rule is

considered to match an input if the input contains all the words of the rule with no restriction on ordering or contiguity. In Figure 3 below, the template matches only input A while the context rule matches A, B, and C. Thus a context rule represents a purely semantic test while a template requires both semantics and syntax (structure).

The process used for matching the input against both templates and context rules is a middle-outward search strategy. That is, the search begins at the ambiguous word and extends outward in both directions. This guarantees finding the resolution key which lies closest to the ambiguous word. This is necessitated for two reasons. First, if an input contains two or more occurences of a particular ambiguous word, each must be paired with its closest resolution key in order to obtain correct results. The examples in Figure 4, though admittedly rather contrived, demonstrate the need for this technique.

B) The Disambiguation Process

The process of disambiguation requires the following elements: a small thesaurus of words needed in the disambiguation process, a set of templates and a set of context rules. The process first reads an input and each word is looked up in the disambiguation thesaurus. Most words are not found and are classified as unknown. The input is first matched against the template set and then the context rule set using the middle-out search strategy. Disambiguation is performed by the first rule successfully matched. The rules in each set are ordered so that the strongest rules, that is the ones that are expected to provide the best disambiguation performance, appear at the top. The weaker or last resort rules appear near the bottom. Scanning the rule list top to bottom matches strong rules weak rules. This critical ordering essentially weights the rules and

Template: COMPUTER PROGRAMMING

Context rule: COMPUTER PROGRAMMING

Inputs: A. Elements of computer programming

B. Programming of digital computers

C. Computer design and programming

Template matches A only

Context rule matches A, B, and C

Comparison of Templates and Context Rules

Figure 3

Input A. It was very cold when he received his college degree.

Action: COLLEGE rather than the temperature reference must be used to disambiguate DEGREE.

Input B. His college degree was to a large degree, well earned.

Action: Each DEGREE must be associated with its nearest resolution key.

Search Strategy (Underline Indicates Resolution Key)

Figure 4

ensures that an input is matched with the rule that has the greatest chance of providing a correct analysis. Associated with each rule is the meaning appropriate to that resolution key. If no match is found between an input and any rule, the ambiguity is considered unresolved. An option may be used in connection with such unresolved inputs. For some ambiguous words one interpretation is much more likely than all the rest. For these a significant saving in the size of the rule sets and in the work involved can be obtained by testing for all but the most likely interpretation. If no matches occur the result is taken by default to be the most likely meaning. This option is used for some of the experiments that follow.

C) Experiments

After classifying the ambiguous words found in the ADI[1] dictionary as true, contextual or syntactic, five true ambiguities are chosen for experimentation. The words are:

DEGREE

TYPE

VOLUME

BOARD

CHARGE

For each word except DEGREE a corpus of 50 sentences is used. A larger corpus is used for DEGREE to provide a more exhaustive test. Each corpus contains all sentences from the ADI documents which contain the ambiguous word as well as other sentences written by the author and other informants. Each corpus is divided into two sets: S-1, called the <u>creation set</u>, and S-2,

---

[1] The ADI Collection is a set of short papers on automation and scientific communication published by the American Documentation Institute, 1963.

called the test set. S-1 contains 20 sentences, S-2 contains the remainder of the corpus. The experimental procedure used for each word is as follows. First, using S-1 only, a thesaurus, template set and context rule set are created by hand. The disambiguation program is then run on S-1. Appropriate additions and modifications are made to the thesaurus and rule sets, and the program is tried again. This continues until the process provides a high degree of success in resolving ambiguities from S-1. The thesaurus and rule sets existing at this point are thus effectively tuned to the creation set S-1. Next, and without further modification of the thesaurus or rules, the disambiguation process is run using S-2 as input. The process is thus tested on an input set it has never seen before, and one to which it is not specifically tuned. The result parameters used are shown in Figure 5 below. Resolution recall indicates what proportion of the total number of ambiguities in the input set are correctly resolved, while resolution precision indicates what proportion of the analyses performed by the system are correct. In order to perform satisfactorily, the process must give reasonably high values for both RR and RP. In the optimal case both values are 1. The results obtained for the five S-2 sets appear in Figure 6 along with totals for all five words. The default option is used in the analysis of TYPE and CHARGE. Inputs for which the system does not perform an analysis for these words are taken to be of a particular interpretation. Thus no inputs are considered unanalyzed (indicated in Figure 6 by an asterisk in the U column).

These results indicate that extended template analysis is a useful and accurate technique for resolution of true ambiguities. The errors which do occur are not, in general, generated by inputs with normal constructions. Rather they are due mostly to idiomatic expressions which are not included in

T	The Total number of ambiguities in the input set. (This number is sometimes larger than the number of sentences in the input set because a few of the sentences contain multiple occurrences of the ambiguous word).
C	The number of ambiguities correctly resolved
I	The number of ambiguities incorrectly resolved.
U	The number of ambiguities not resolved in any way.
RR	Resolution Recall = C/T
RP	Resolution precision = C/(C+I)

Result Parameters

Figure 5

WORD	T	C	I	U	RR	RP
DEGREE	92	84	4	4	.92	.93
TYPE	30	29	1	*	.97	.97
VOLUME	30	27	1	2	.90	.96
BOARD	30	22	0	8	.73	1.00
CHARGE	32	30	2	*	.94	.94
TOTAL	214	192	8	14	.90	.96

* indicates default used

Results of Disambiguation of S-2 Sets

Figure 6

in the creation set. As an example the expression ON BOARD is not in S-1 for BOARD. This in turn leads to a number of inputs in the test set being unanalyzed. While such idioms in natural language may prevent perfect disambiguation quality, they occur relatively infrequently in practice and thus reduce the system performance only slightly.

D) Further Disambiguation Processes

A number of further processes are suggested by the experiments performed here. First, a statistical weighting can be attached to each resolution. This would represent the probability of correctness of the given rule. The context of the ambiguous word could then be searched for all, not just one, resolution key. For each key found, a correlation is calculated which takes into account the probability of the rule being correct as well as the key's distance from the ambiguous word. The rule with the highest correlation is then used. In this way a strong resolution key can take precedence over a weak one lying closer to the ambiguous word.

A second addition is the use of a variable context. All methods for disambiguation presented here including those by Borillo and Virbel and template analysis use a fixed context size for all words. However the optimal context size varies from word to word. It would thus be better to associate with each word, the context width that works. A third possible future technique is to use antirules. These are rules which if matched, tell what interpretation of the ambiguous words cannot be used. For example, if Y appears in an input, interpretation X is prohibited even if indicators for X are present. These extensions, however, are beyond both the scope and the and the spirit of the present study.

5. Learning to Disambiguate Automatically

A) Introduction

The processes of creating and modifying the sets of templates and context rules as presented in section 4 are relatively straightforward and algorithmic in nature. Rules are constructed from creation set inputs by fairly specific means. Likewise, in rule modification an erroneous rule is removed and replaced by one or more rules which perform correctly. It seems possible that these tasks can be handled by computer. Thus instead of telling the program what to do by manually supplying rules, the system would learn to disambiguate by creating and modifying its own rule sets. The advantages of such a system over one of the type described previously are obvious. First, it eliminates the need for a human analyst to study sample inputs and create template and context rule sets. Second, the system is not static. By learning from inputs and its own mistakes it is constantly improving its performance. This process can even be used to tailor a system to an individual user. Disambiguation rules, or rules for any number of other processes, that are designed by or for a particular user are not always well suited for others. By allowing the system to learn separately from each individual, the particular needs of each user are satisfied. This section discusses some techniques for automatically learning to disambiguate.

B) Dictionary and Corpus

When disambiguation rules are prepared by hand, the words which are to be used in the disambiguation are known in advance. The disambiguation dictionary need only contain these relevant words and thus is quite small. In the learning process, there is no prior knowledge of the words that are to be used to facilitate disambiguation. For this reason a full dictionary

containing all the words in the input must be employed initially. This large dictionary, however, is needed only temporarily. After the initial instability of the learning process has settled down and relatively fixed rule sets remain, the words in these rule sets may be used to construct a small disambiguation dictionary which can be used thereafter.

The corpora used in this study are very special. In practice an operational learning system has a very large input set. The learning process may thus extend over hundreds or even thousands of inputs. However, such large data sets are neither readily available nor practical for an experimental system. For this reason it is necessary to develop a small corpus which simulates a much larger one. This is a technique used in a number of experimental studies including Harris' investigation of morpheme boundaries [5]. The rules governing this stem from two fundamental maxims of education. First, a student or learning device cannot be expected to answer a question about something he has not seen previously. That is, a student's first exposure to a concept must be in a learning not a testing environment. And second, to evaluate learning quality, testing is required. Basically these rules say that to test properly a learning system, each concept to be learned must occur at least twice in the input, once for learning and subsequently for testing. Single occurrences are undesirable because if they are considered as a test, they violate the first rule, while if, as the first rule stipulates, the single occurrence is considered for learning only, no testing can occur and the second rule is violated. Large data collections are likely to have multiple occurrences of most concepts. This however is not true for small corpora; and care must be taken to ensure such repetition. To accomplish this the following algorithm is used for corpus construction for each ambiguous word. First, a set of 20 short sentences is written, each containing

the ambiguous word. No restriction on vocabulary or construction is imposed for these first 20 sentences. Next, 40 more sentences are written using only words found in the first 20. Again no restriction on construction is imposed. The resulting 60 sentences are sufficiently restricted in vocabulary to ensure that most words and constructs occur at least twice. The corpus thus simulates a corner of a much larger collection. To determine if the system is unlearning previously learned information while learning new material, the actual input consists of the set of 60 sentences repeated three times. Each set of 60 is randomly permuted to eliminate any prejudice due to ordering. The input format is summarized in Figure 7. Such corpora currently exist for three ambiguous words:

DEGREE

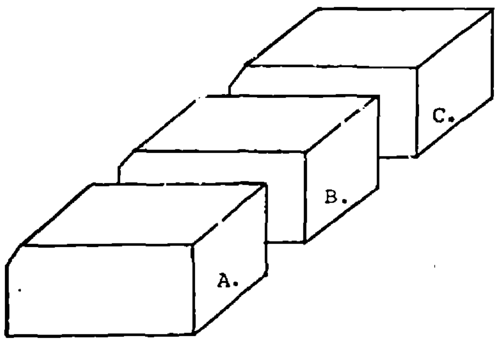
TYPE

VOLUME

These are chosen from the set used in previous experiments because VOLUME is rather difficult to disambiguate, TYPE is fairly easy, and DEGREE is between, tending toward difficulty. It is felt that the results obtained and the problems encountered with these words are typical of those to be expected for most other words.

C) The Learning Process

The learning process is implemented as a set of subroutines to the system described in section 4. Dynamic template and context rule lists replace the fixed sets. Initially there are no rules in these sets. The processing of each input sentence proceeds as follows. After the input is



A: Corpus, permutation 1

B: Corpus, permutation 2

C: Corpus, permutation 3

Summary of Input Format for Learning System

Figure 7

read and the ambiguity located, the system attempts to disambiguate the word using templates and context rules currently in the system. When the analysis is complete, the system looks at the correct answer. If the analysis is correct, the system is assumed to contain the appropriate rules for the recognition of the input structure and the system goes on to the next input. If the system is unable to resolve the ambiguity, that is, if no existing rule matches the input, new rules must be added. New templates and context rules are created using the prespecified parameters I and J. I specifies the size of the area around the ambiguous word from which templates are to be made. Similarly J indicates the size of the area from which context rules are to be made. In general J is larger than I since unstructured resolution keys can lie farther away from the ambiguous word than do structured keys. For this study I and J have the values of 2 and 5 respectively. A template is made for each word of the input sentence which lies within plus or minus I of the ambiguous word. The templates preserve the ordering and the relative distance between words. A context rule is created for each word within plus or minus J of the ambiguous word provided the word is not found on a predefined exclusion list. As indicated previously, context rules have no ordering or contiguity restriction. The exclusion list contains words which are of no value in establishing context. These include articles, some prepositions, forms of the verb TO BE, etc. The list is created by consideration of context in general and without any reference to specific words being disambiguated. The exclusion list is not used in the creation of templates because some apparently trivial words are actually important when found in particular structural relationships to an ambiguous word. For example, one of the primary templates for the disambiguation of TYPE is

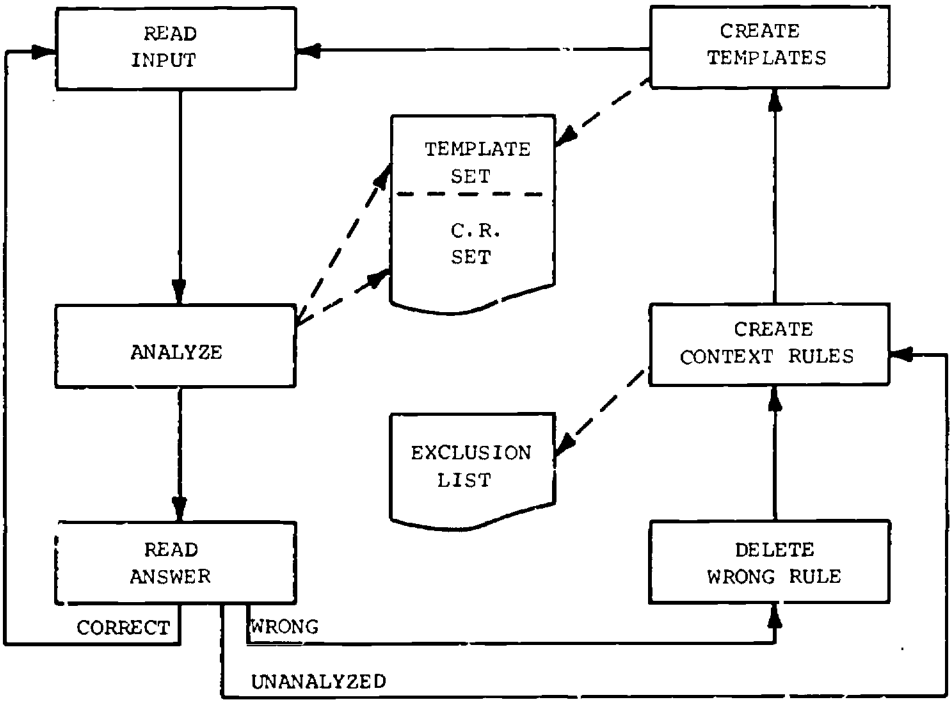
TYPE OF

The templates and context rules created by this process are first placed in a temporary store and checked against rules already in the permanent template and context rule sets. All rules in the temporary store which are not duplicates of existing rules are added to the bottom of the appropriate permanent set. This completes the action for an unanalyzed input.

The third possible outcome is for the system to produce an erroneous analysis. In this case the rule sets not only lack the rules needed for correct analysis, but also contain an erroneous rule. Therefore when this situation arises, the rule which produces the incorrect result must first be removed from the rule set. Each rule lying below the deleted rule is then popped up one position in the rule list. Next, templates and context rules are added just as in the previous case. The operation is summarized in Figure 8.

Critical ordering of rules, as is done in section 4 is not possible when rules are created automatically. However the process of deleting a rule and popping up those below it and then adding the new rules at the bottom tends to make the better rules, that is those which do not get deleted, filter to the top. While this method may not be as effective as critical ordering by hand, it does tend to concentrate the better rules near the top of the lists. The top down search strategy thus matches rules against an input in roughly best first order. Experimental results which verify this are presented later.

Ideally, a system such as that described above operating on a corpus of the form shown in Figure 7 should generate the following type of results. The first few inputs are of course unanalyzed due to the lack of information. As more inputs are read, the overall system performance should begin a steady ement. Eventually the system should stabilize with a fixed rule set



Summary of Learning Process

Figure 8

and near perfect disambiguation. From this point the system should never unlearn. That is, it should never err on an input that it previously analyzed correctly. Likewise it should not be overly sensitive to the order in which inputs are introduced. Actual experimental results obtained compare quite favorably with this idealized behavior. These results are presented in subsection E.

D) Spurious Rules

The learning process presented in part B has a few inherent problems. These center mainly around the treatment of spurious rules. A spurious rule is defined to be a template or context rule which does not discriminate between interpretations of an ambiguous word. As an example, assume that templates and context rules for disambiguation of TYPE are made from input A in Figure 9. One of the templates extracted from this input is LARGE TYPE. This however is of no value as can be seen from input B. Thus LARGE TYPE is considered a spurious rule.

Input A: The book is printed in large type.
(interpretation 1, "printing")

Input B: A tiger is a large type of cat.
(interpretation 2, "kind or variety")

Example of a Spurious Rule

Figure 9

The difficulty with the process as presented in subsection C (to be called version 1 in the remainder of this study), can be visualized by the

following example. Assume rules are learned from input A in Figure 9. Included among these is the spurious rule LARGE TYPE which is associated with interpretation 1. Assume also that input B is then processed by a match with LARGE TYPE and hence incorrectly associated with interpretation 1. Version 1 then deletes the interpretation 1 template and substitutes one which is identical except for its association with interpretation 2. Thus a spurious rule is deleted but replaced with one equally spurious. This actually produces a slight improvement since the new rule is inserted at the bottom of the list and thus is less likely to be matched than the one it replaces. But the spurious rules remain and can cause further errors. They may even cause a thrashing back and forth between interpretations and thus prevent stability.

One possible solution to this is implemented in version 2. Whenever a rule is to be deleted because it causes an incorrect analysis, the set of new incoming rules is checked for an occurrence of this same rule. If found, the matched rule is not added to the permanent rule set. Thus using version 2, the incorrect analysis of input B would not only remove LARGE TYPE from the template set but would also prevent this same template (with a different interpretation) from entering the set at that time. In the short run this has the effect of eliminating spurious rules from the system. But since no record is kept, these same spurious rules may reenter the system the next time they occur. A reoccurrence of input A follwoing input B for example, would put LARGE TYPE back on the rule list. Thus while version 2 does provide some advantages over version 1, there is still room for improvement.

The second modification, version 3, solves the difficulty inherent version 2. When spurious rules are located, they are removed from

both the rule set and the new entering set as in version 2. But in addition the rule is recorded on a list of undesirable rules. All incoming rules are checked against the undesirable list. If a match is found, that incoming rule is deleted. In this way a spurious rule, once found, is permanently prevented from reentering the system. While this process may cause a mild retardation in the learning rate due to the decreased number of rules used, the slowdown is more than compensated by the increased accuracy of the results. The workings of versions 1, 2, and 3 are summarized in Figure 10.

E) Experiments and Results

The experimentation consists of processing each of the three corpora with the three system versions, a total of nine runs in all. The corpora are each 180 sentences in length and are described previously in subsection B. The performance measures that are taken are shown in Figure 11. These results are tabulated in Figure 12. Figure 13 shows the resolution recall and precision for each word calculated at ten document intervals. Averages for the results in Figure 13 are presented in Figure 14. These results show how the overall system performance improves as more inputs are seen, thus indicating a true learning process. These charts also show the general superiority of version 3 over the other two. To indicate this fact more clearly, Figure 15 shows the difference in resolution recall and precision for the three versions averaged over all corpora. Version 1 is taken as the standard and lies on the x axis. Displacement above or below the x axis represents superiority or inferiority relative to version 1. These graphs show that version 2 and especially version 3 improve both resolution recall and precision over version 1. That is, not only do they perform more correct analyses than version 1, they also perform fewer incorrect analyses. Usually

INPUT	STATUS AFTER INPUT			
	V-1 Rule Set*	V-2 Rule Set*	V- 3 Rule Set	V- 3 Undesirable Rule List
A	LARGE TYPE (1)**	LARGE TYPE (1)	LARGE TYPE (1)	-
B	LARGE TYPE (2)**	-	-	LARGE TYPE
A	LARGE TYPE	LARGE TYPE	-	LARGE TYPE
	(1)	(1)	-	LARGE TYPE
	(1)	(1)		

* This chart shows only the part of the rule set that is relevant to this discussion.

** Numbers in parentheses indicate the interpretation associated with the rule.
Interpretation 1 is printing
Interpretation 2 is kind or variety

Summary of Versions 1, 2, and 3

Figure 10

T	The total number of ambiguities in the data set
C	The number of correctly resolved ambiguities
I	The number of incorrectly resolved ambiguities
U	The number of unresolved ambiguities
RR	Resolution Recall = C/T
RP	Resolution Precision = C/(C+I)

Performance Measures

Figure 11

WORD	VERSION	T	C	I	U	RR	RP
DEGREE	1	180	155	19	6	.86	.89
DEGREE	2	180	158	14	8	.88	.91
DEGREE	3	180	160	12	8	.89	.93
TYPE	1	180	166	10	4	.92	.94
TYPE	2	180	166	7	7	.92	.96
TYPE	3	180	164	4	12	.91	.98
VOLUME	1	180	144	30	6	.80	.83
VOLUME	2	180	144	30	6	.80	.83
VOLUME	3	180	152	15	13	.84	.91
TOTALS	1	540	465	59	16	.86	.89
	2	540	468	51	21	.87	.90
	3	540	476	31	33	.88	.94

General Results of Learning Process

Figure 12

DEGREE NO. OF INPUTS PROCESSED	VERSION 1		VERSION 2		VERSION 3	
	RR	RP	RR	RP	RR	RP
10	.40	.60	.40	.80	.40	.80
20	.55	.79	.50	.77	.50	.77
30	.60	.75	.57	.77	.57	.77
40	.67	.79	.65	.81	.65	.81
50	.64	.72	.66	.79	.66	.79
60	.66	.74	.68	.79	.70	.81
70	.70	.77	.71	.81	.73	.82
80	.71	.77	.74	.82	.75	.83
90	.73	.79	.77	.84	.78	.85
100	.76	.81	.79	.86	.80	.87
110	.78	.83	.80	.86	.82	.88
120	.80	.84	.82	.87	.83	.89
130	.82	.85	.83	.89	.85	.90
140	.83	.86	.84	.89	.86	.91
150	.83	.87	.85	.90	.87	.92
160	.84	.88	.86	.91	.87	.92
170	.85	.88	.87	.91	88	.93
180	.86	.89	.88	.92	.89	.93

Recall and Precision Results at Ten Input Intervals

Ambiguous word is DEGREE

Figure 13A

TYPE NO. OF INPUTS PROCESSED	VERSION 1		VERSION 2		VERSION 3	
	RR	RP	RR	RP	RR	RP
10	.50	.71	.50	.71	.50	.71
20	.65	.76	.65	.81	.65	.81
30	.67	.77	.67	.83	.70	.88
40	.72	.81	.73	.85	.75	.88
50	.78	.85	.78	.89	.76	.90
60	.82	.88	.82	.91	.80	.92
70	.84	.89	.84	.92	.83	.94
80	.86	.91	.86	.93	.85	.94
90	.88	.92	.88	.94	.84	.95
100	.89	.93	.89	.95	.86	.96
110	.89	.92	.89	.94	.87	.96
120	.89	.92	.90	.95	.88	.96
130	.90	.93	.90	.95	.88	.97
140	.91	.93	.91	.95	.89	.97
150	.91	.94	.91	.96	.89	.97
160	.91	.94	.91	.95	.90	.97
170	.92	.94	.92	.96	.91	.97
180	.92	.94	.92	.96	.91	.98

Recall and Precision Results at Ten Input Intervals

Ambiguous word is <u>TYPE</u>

Figure 13B

VOLUME NO. OF INPUTS PROCESSED	VERSION 1		VERSION 2		VERSION 3	
	RR	RP	RR	RP	RR	RP
10	.10	.17	.20	.33	.20	.33
20	.30	.40	.35	.47	.45	.64
30	.40	.50	.43	.54	.53	.70
40	.47	.56	.50	.59	.55	.67
50	.54	.61	.54	.61	.60	.71
60	.58	.65	.60	.67	.65	.75
70	.61	.67	.63	.69	.69	.79
80	.65	.70	.65	.70	.73	.82
90	.68	.73	.68	.73	.76	.84
100	.71	.76	.70	.74	.78	.86
110	.73	.77	.72	.76	.80	.87
120	.74	.78	.73	.77	.79	.87
130	.76	.80	.75	.78	.80	.88
140	.77	.81	.76	.80	.81	.89
150	.78	.81	.77	.81	.83	.90
160	.79	.82	.79	.82	.83	.90
170	.79	.82	.79	.82	.84	.90
180	.80	.83	.80	.83	.84	.91

Recall and Precision Resutls at Ten Input Intervals

Ambiguous word is VOLUME

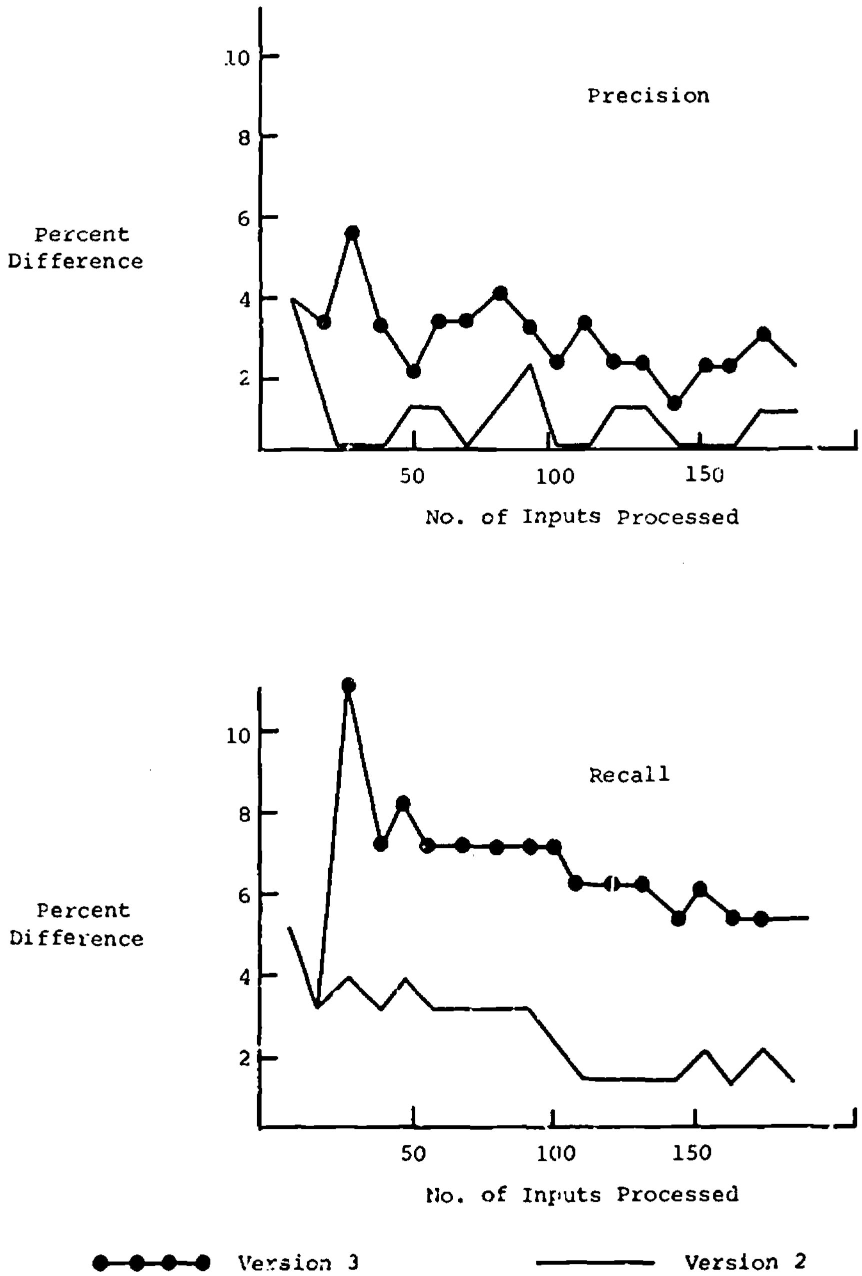
Figure 13C

AVERAGES NO. OF INPUTS PROCESSED	VERSION 1		VERSION 2		VERSION 3	
	RR	RP	RR	RP	RR	RP
10	.33	.56	.37	.61	.37	.61
20	.50	.65	.50	.68	.53	.74
30	.55	.67	.55	.71	.60	.78
40	.62	.72	.62	.75	.65	.79
50	.65	.72	.66	.76	.67	.80
60	.69	.76	.70	.79	.72	.83
70	.72	.78	.72	.81	.75	.85
80	.74	.79	.75	.82	.78	.86
90	.76	.81	.78	.84	.79	.88
100	.79	.83	.79	.85	.81	.90
110	.80	.84	.80	.85	.83	.90
120	.81	.85	.82	.86	.83	.91
130	.82	.86	.83	.87	.84	.92
140	.84	.87	.84	.88	.85	.92
150	.84	.87	.84	.89	.86	.93
160	.85	.88	.85	.89	.87	.93
170	.85	.88	.86	.90	.88	.93
180	.86	.89	.87	.90	.88	.94

Average Recall and Precision for All Corpora

Tabulated at Ten Input Intervals

Figure 14



Average Improvement Achieved by Versions 2 and 3
Over Version 1

Figure 15

this results in an increased number of unanalyzed inputs. This is actually a very desirable result since if a choice must be made between an input being analyzed incorrectly or not analyzed at all, the latter seems preferable. An example of this can be seen in Figure 13B. Version 2 produces only a few more correct analyses than does version 1, and thus the recall results show very little difference. However version 2 produces many fewer incorrect analyses thus significantly improving the precision results.

The results shown so far are prejudiced downward by the inclusion of the start-up portion of the learning process which necessarily performs poorly. Therefore a more important measure of system performance is a moving average. Figure 16 shows for each word the number of disambiguations performed correctly, incorrectly, and unanalyzed for each ten sentence group. These charts clearly indicate the anticipated poor start, the gradual improvement, and the final stabilization at near perfect performance. A 10 in the "Correct" column represents perfect resolution for that sentence group. These statistics are summarized by Figure 17. And in Figure 18, these averages are shown graphically. The x axis is the interval number. Interval 5, for example, contains inputs 41-50, etc. The y axis represents the number of correct analyses out of a possible 10. These charts are very graphic proof that the learning process builds and stabilizes at a high performance level.

Several other statistics are worthy of note. Figure 19 shows for each run the number of spurious templates and context rules contained in the rule sets at the end of that run. This number is broken down to show how many of these spurious rules occur in the first, middle, and last third of their respective rule sets. These figures indicate first that most rules learned by the system are not spurious; and secondly, that spurious rules

DEGREE

INPUTS	VERSION 1			VERSION 2			VERSION 3		
	C	I	U	C	I	U	C	I	U
1-10	4	1	5	4	1	5	4	1	5
11-20	7	2	1	6	2	2	6	2	2
21-30	7	3	0	7	2	1	7	2	1
31-40	9	1	0	9	1	0	9	1	0
41-50	5	5	0	7	3	0	7	3	0
51-60	8	2	0	8	2	0	9	1	0
61-70	9	1	0	9	1	0	9	1	0
71-80	8	2	0	9	1	0	9	1	0
81-90	9	1	0	10	0	0	10	0	0
91-100	10	0	0	10	0	0	10	0	0
101-110	10	0	0	9	1	0	10	0	0
111-120	10	0	0	10	0	0	10	0	0
121-130	10	0	0	10	0	0	10	0	0
131-140	10	0	0	10	0	0	10	0	0
141-150	9	1	0	10	0	0	10	0	0
151-160	10	0	0	10	0	0	10	0	0
161-170	10	0	0	10	0	0	10	0	0
171-180	10	0	0	10	0	0	10	0	0

C No. of Correct Analyses out of a Possible 10
I No. of Incorrect Analyses
U No. Unanalyzed

Disambiguation Performance for Ten Input Groups

Ambiguous word is DEGREE

Figure 107

TYPE INPUTS	VERSION 1			VERSION 2			VERSION 3		
	C	I	U	C	I	U	C	I	U
1-10	5	2	3	5	2	3	5	2	3
11-20	8	2	0	8	1	1	8	1	1
21-30	7	2	1	7	1	2	8	0	2
31-40	9	1	0	9	1	0	9	1	0
41-50	10	0	0	10	0	0	8	0	2
51-60	10	0	0	10	0	0	10	0	0
61-70	10	0	0	10	0	0	10	0	0
71-80	10	0	0	10	0	0	10	0	0
81-90	10	0	0	10	0	0	8	0	2
91-100	10	0	0	10	0	0	10	0	0
101-110	9	1	0	9	1	0	10	0	0
111-120	9	1	0	10	0	0	10	0	0
121-130	10	0	0	9	0	1	8	0	2
131-140	10	0	0	10	0	0	10	0	0
141-150	10	0	0	10	0	0	10	0	0
151-160	9	1	0	9	1	0	10	0	0
161-170	10	0	0	10	0	0	10	0	0
171-180	10	0	0	10	0	0	10	0	0

C No. of Correct Analyses Out of a Possible 10
I No. of Incorrect Analyses
U No. Unanalyzed

Disambiguation Performance for Ten Input Groups

Ambiguous word is TYPE

Figure 16B

VOLUME

INPUTS	VERSION 1			VERSION 2			VERSION 3		
	C	I	U	C	I	U	C	I	U
1-10	1	5	4	2	4	4	2	4	4
11-20	5	4	1	5	4	1	7	1	2
21-30	6	3	1	6	3	1	7	2	1
31-40	7	3	0	7	3	0	6	4	0
41-50	8	2	0	7	3	0	8	1	1
51-60	8	2	0	9	1	0	9	1	0
61-70	8	2	0	8	2	0	9	0	1
71-80	9	1	0	8	2	0	10	0	0
81-90	9	1	0	9	1	0	10	0	0
91-100	10	0	0	9	1	0	10	0	0
101-110	9	1	0	9	1	0	10	0	0
111-120	9	1	0	9	1	0	7	1	2
121-130	10	0	0	9	1	0	9	0	1
131-140	9	1	0	10	0	0	10	0	0
141-150	9	1	0	9	1	0	10	0	0
151-160	9	1	0	10	0	0	9	0	1
161-170	9	1	0	9	1	0	9	1	0
171-180	9	1	0	9	1	0	10	0	0

C No. of Correct Analyses out of a Possible 10
I No. of Incorrect Analyses
U No. Unanalyzed

Disambiguation Performance for Ten Input Groups

Ambiguous word is VOLUME

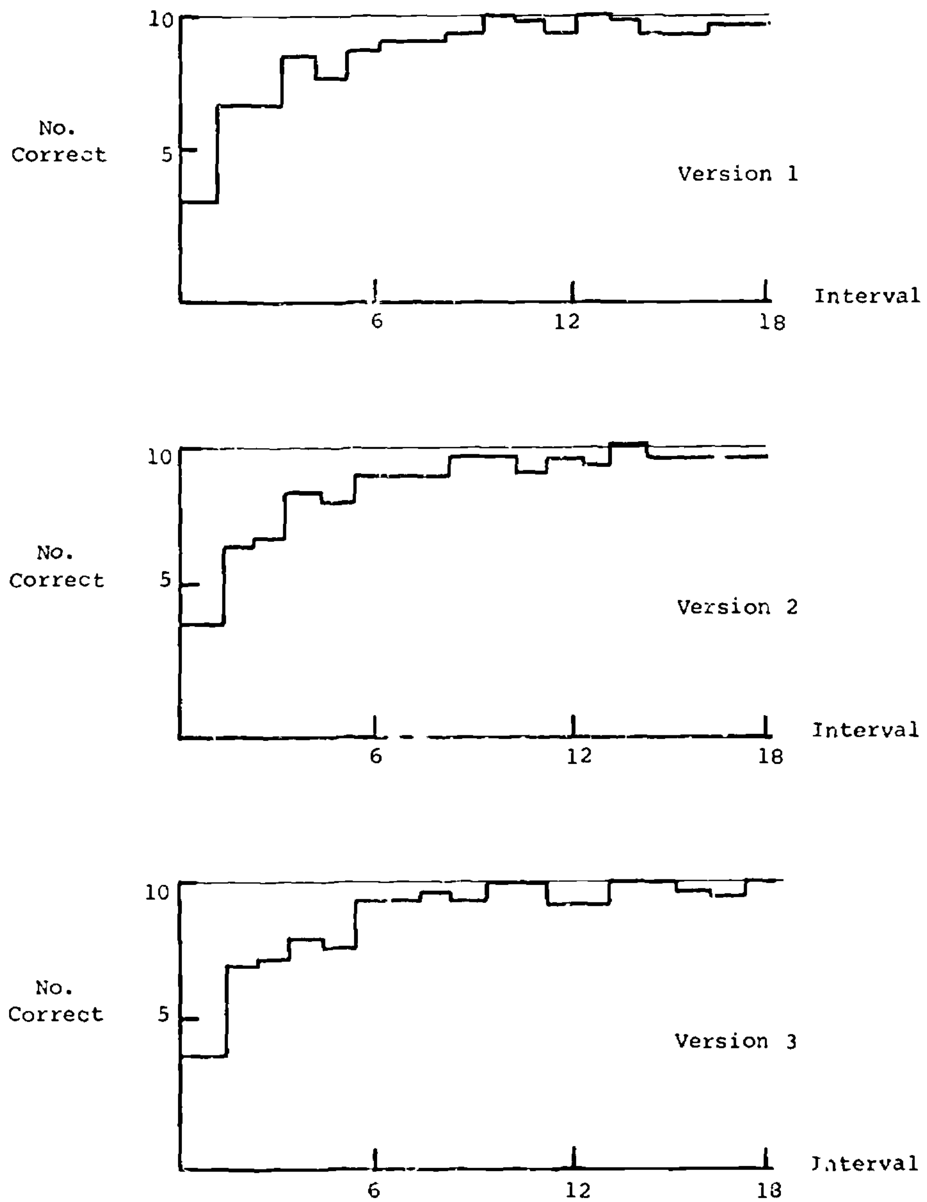
Figure 16C

INPUTS	AVERAGE NO. OF CORRECT ANALYSES OUT OF POSSIBLE 10		
	VERSION 1	VERSION 2	VERSION 3
1-10	3.33	3.67	3.67
11-20	6.67	6.67	7.00
21-30	6.67	6.67	7.33
31-40	8.33	9.33	8.00
41-50	7.67	8.00	7.67
51-60	8.67	9.00	9.33
61-70	9.00	9.00	9.33
71-80	9.00	9.00	9.67
81-90	9.33	9.67	9.33
91-100	10.00	9.67	10.00
101-110	9.97	9.00	10.00
111-120	9.33	9.67	9.00
121-130	10.00	9.33	9.00
131-140	9.97	10.00	10.00
141-150	9.33	9.67	10.00
151-160	9.33	9.67	9.97
161-170	9.67	9.67	9.67
171-180	9.67	9.67	10.00

Average Number of Correct Analyses for Each Ten Input Group

Maximum is 10

Figure 17



Average Number of Correct Analyses
For Each Ten Input Group

Figure 18

RUNS	# OF TEMPS	SPURIOUS				# OF C.R.	SPURIOUS			
		1/3	2/3	3/3	TOT		1/3	2/3	3/3	TOT
DEGREE V-1	31	2	2	7	11	25	0	3	4	7
DEGREE V-2	28	2	1	5	8	23	0	3	3	6
DEGREE V-3	19	1	0	1	2	22	0	1	5	6
TYPE V-1	21	1	3	4	8	15	0	3	3	6
TYPE V-2	18	1	3	2	6	12	0	2	1	3
TYPE V-3	20	1	3	3	7	10	0	1	1	2
VOLUME V-1	36	4	6	9	19	22	0	2	3	5
VOLUME V-2	31	3	3	8	14	21	0	2	2	4
VOLUME V-3	25	2	3	5	10	18	0	1	3	4
TOTAL	229	17	24	44	85	168	0	18	25	43

Number of Spurious Rules Found in the First, Middle and Last Third of Each Rule Set (For Each Run).

Figure 19

tend to be densest at the bottom of the rule sets. Thus due to the top-down search strategy, correct rules are far more likely to be chosen than spurious ones.

As stated previously one requirement for a good learning system is that it not be prone to unlearning. An input is considered to be unlearned if it is seen once and analyzed correctly and subsequently seen again and analyzed incorrectly. Figure 20 shows the number of unlearned inputs for each of the nine experimental runs. The low values here clearly indicate that once the system has learned to disambiguate a particular input, that capability remains learned. Also, the fact that versions 2 and 3 perform better than version 1 with respect to unlearning indicates that the prevention of spurious rules is an aid in the prevention of unlearning. Unlearning may stem from sources other than the system itself. If a user provides incorrect information to a learning system, improper rules and subsequent unlearning may result. In an operational learning system it may therefore be necessary for an analyst to review periodically the newly learned rules prior to their final acceptance into the permanent rule set.

One final investigation is to look at the contents of the undesirable rule lists following each version 3 run. Figure 21 shows the total number of rules in the lists and the number which by hand analysis are found to be actually spurious. Ideally all rules in these lists should be spurious; and the figures shown are quite close to this ideal. These results show that the system is able to learn not only the rules which make good disambiguaters, but also those which are not useful. The results presented here show these processes are truly capable of learning to disambiguate with a high degree of success.

F) Extensions

There are numerous other applications for a learning technique such

WORD	VERSION 1	VERSION 2	VERSION 3
DEGREE	1	1	0
TYPE	1	1	0
VOLUME	4	3	2
AVERAGE	2	1.67	0.67

Number of Unlearned Inputs for Each Run

Figure 20

RUN	USELESS TEMPLATE LIST		USELESS C.R. LIST	
	LENGTH	# SPUR	LENGTH	# SPUR
DEGREE	10	9	2	2
TYPE	1	1	1	1
VOLUME	9	8	3	3
TOTAL	20	18	6	6
ACCURACY	90%		100%	

Composition of the Useless Rule Lists

(Version 3 Only)

Figure 21

as the one presented previously. A large system with many users may be able to learn the individual needs and techniques of its users. The system could thus tailor a specialized subsystem to each individual. In the area of information retrieval a system might be able to learn to modify techniques and parameters in order to improve relevance feedback performance for a particular collection and user. In nearly any application where a set of rules or parameters must be created in order to perform some form of analysis, the learning technique is potentially valuable, especially where many such sets must be created to meet the needs of many users.

The learning process can also be applied to natural language analysis in the resolution of pronouns. Unlike ambiguities which have multiple meanings, pronouns have no meaning in isolation. To determine meaning, the word to which the pronoun refers must be located. This could be accomplished in the following way. The learning process looks at each noun in the vicinity of the pronoun and learns their contexts. These are then compared with the context of the pronoun and the noun with the best match used. There are of course some problems to be solved. For example, not all pronouns refer to a specific thing. The fact that some pronouns encompass large concepts or merely provide an impersonal subject can be seen in the second and third example sentences below.

A. Take an egg and break <u>it</u> into a bowl.
   (specific reference)

B. The consequence of <u>this</u> is that the project is feasible.
   (multiple reference)

C. These results show that <u>it</u> is possible.
   (impersonal)

can provide a more accurate natural language analysis process and

improve performance in any natural language application.

6. Conclusion

This study is intended first to demonstrate the importance of disambiguation in various forms of natural language analysis, and to motivate investigation into the automation of this process. It also serves as a test of the template analysis facility. The study shows that it is possible to perform this disambiguation with a high degree of accuracy using an extended form of template analysis and a predetermined set of structured templates and unstructured context rules. The creation of these rules requires an analyst to examine typical inputs and determine the words or structures which indicate the intended meaning of the ambiguous word. As is shown in 5 this manual process may be eliminated by implementation of a process which allows the system to disambiguate for itself. With the exception of the first few inputs for which the performance is understandably low, the learning process demonstrates the same high degree of accuracy achieved with the hand made disambiguation rules. Not only is the system able to learn which rules provide good disambiguation, it can also determine which rules do not, and exclude these rules from the system. The learning process has applications in many areas and template analysis appears sufficiently general to facilitate many of the applications.

References

[1] Borillo, A., and Virbel, J., Resolution des Polysemies Dans L' Indexation Automatique de Resumes, Centre National de la Recherche Scientifique, Section D' Automatique Documentaire, June, 1966.

[2] Coyaud, M., Resolution of Lexical Ambiguities in Ophthalmology, Cornell University, Ithaca, New York, 1968.

[3] Dimsdale, B., and Lamson, B. G., A Natural Language Information Retrieval System, Proceedings of the IEEE, Vol. 54, No. 12, December 1966.

[4] Douglas, E., Mandersloot, W., and Spicer, N., Thesaurus Control -- the Selection, Grouping and Cross-referencing of Terms for Inclusion in a Coordinate Index Word List., Journal of the American Society for Information Science, January - February, 1970.

[5] Harris, Z. S., From Phoneme to Morpheme, Language, Vol. 31, No. 2, 1955.

[6] Kellogg, C. H., A Natural Language Compiler for On-line Data Management, AFIPS Conference Proceedings, Vol. 33, Proc. AFIPS 1968 Fall Joint Computer Conference, Vol. 33, Thompson Book Company, Washington, D. C.

[7] Lesk, M. E., A Proposal for English Text Analysis, Bell Telephone Laboratories, 1969.

[8] Moyne, J. A., A Progress Report on the Use of English in Information Retrieval, IBM Corporation, Federal Systems Center, Gaithersburg, Maryland June 1969.

[9] Quillian, M. R., The Teachable Language Comprehender: A Simulation Program and Theory of Language, CACM Vol. 12, No. 8, August 1969.

[10] Rangathan, S. R., Recall Value and Entry Word in Headings, Library Science, Vol. 6, No. 4, December 1969.

[11] Simmons, R. F., Answering English Questions by Computer: A Survey, CACM, Vol. 8, No. 1, January 1965.

[12] Simmons, R. F., Synthex. In Orr, W.D., (Ed.), Conversational Computers, John Wiley and Sons, Inc., New York, 1968.

[13] Simmons, R. F., Natural Language Question-answer Systems: 1969, CACM, Vol. 13, No. 1, January 1969.

[14] Stone, P. J., Improved Quality of Content Computerized-disambiguation Rules for High Frequency English Words. In Analysis of Communication Content, Wiley, New York, 1969.

[15] Thompson, F. B. DEACON Type Query Systems. In Orr, W. D., (Ed.), Conversational Computers, John Wiley and Sons, Inc., New York, 1968.

[16] Weiss, S. F., A Template Approach to Natural Language Analysis for Information Retrieval, Ph.D. Thesis, Cornell University, Ithaca, New York, 1970.