VIII. An Experimental Investigation of Automatic Hierarchy Generation

G. Blomgren, A. Goodman, and L. Kelly

Abstract

In automatic or semi-automatic document retrieval systems, a hierarchical arrangement of concepts or terms affords modification of a query in three ways : generalization, specialization, or expansion with synonyms. Hierarchies are usually constructed manually. A method for automatic generation of hierarchies is proposed, and experimental results are presented.

1. Introduction

An automatic or semi-automatic document retrieval system usually includes a thesaurus of concepts or terms which is used to expand queries. For a given query, thesaurus entries which are similar to terms in the query are added to its vector of terms. The search for relevant documents then continues with the expanded query [5,6,7].

Some systems, such as SMART at Harvard, employ a hierarchical arrangement of concepts or terms to modify queries.[3] Such an arrangement connects concepts by "parent-son" and "brother-brother" relationships. A parent concept is more general than its sons; brothers share an equivalent ranking. Thus a query may be generalized by adding to its vector of terms the parents of those terms; contrariwise, a query may be specialized by adding the sons of its terms. The addition of brothers represents inclusion of similar terms.[5]

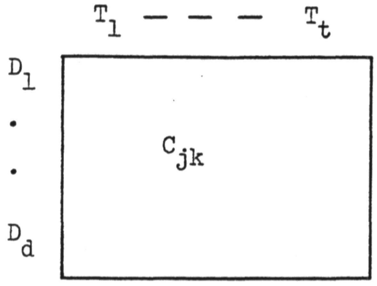
Hierarchies of concepts or terms are usually prepared manually from the documents in a particular collection. Such a preparation requires much time and involves human judgment of relationships between concepts. Human judgment is likely to vary substantially from person to person, resulting in various hierarchies from the same document collection; moreover, these judgments rely on knowledge and experience external (and perhaps extraneous) to the collection. Analogous problems arise in manual indexing of documents or abstracts.

These delays and inequities of manual construction might be overcome by an automatic scheme implemented on a computer. Such a scheme offers two advantages :

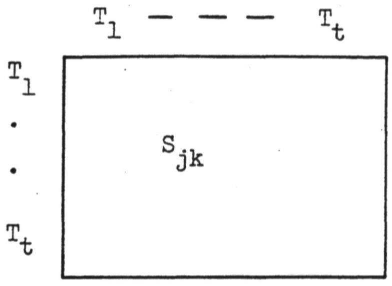
1) Machine preparation eliminates the time-consuming, routine work in outlining a hierarchy.
2) In making decisions about concept relationships, the machine depends only upon the particular documents in the collection, avoiding extraneous information.[2,8]

2. Automatic Construction of Hierarchies

The basic source of information about relationships between terms is the document-term matrix, a listing of documents showing the degree of relevance of each term to each document. The first step is the construction of a non-symmetrical term-term matrix. The authors use the following algorithm [7] :



Document-term



Term-term

where

$$S_{jk} = \frac{\sum\limits_{i=1}^{d} \text{MIN}\,(C_{ij},C_{ik})}{\sum\limits_{i=1}^{d} C_{ij}} \qquad S_{kj} = \frac{\sum\limits_{i=1}^{d} \text{MIN}\,(C_{ij},C_{ik})}{\sum\limits_{i=1}^{d} C_{ik}} \quad \text{for } j\neq k$$

$$0 \leq S_{jk} \leq 1 \quad \text{for} \quad j\neq k$$

S_{jj} is not defined and is never used

$$S_{jk} \cdot \sum_{i=1}^{d} C_{ij} = S_{kj} \cdot \sum_{i=1}^{d} C_{ik}$$

so that

$$S_{kj} = S_{jk} \cdot \frac{\Sigma\, C_{ij}}{\Sigma\, C_{ik}}$$

The second step is the evaluation of relationships between pairs of terms. Choosing a "cutoff" parameter $0 \leq K \leq 1$, apply the following rules [7]:

1) $S_{jk} < K$, $S_{kj} < K$ — T_j and T_k are <u>unrelated</u>, since the two terms generally are not relevant to the same documents.

2) $S_{jk} \geq K$, $S_{kj} \geq K$ — T_j and T_k are <u>similar</u>, since both terms generally are relevant to the same documents. Similar terms are called <u>brothers</u>.

3) $S_{jk} \geq K$, $S_{kj} < K$ T_k is a parent of T_j, since T_j and T_k appear together often, but T_k is relevant to more documents. (Or, T_j is a son of T_k.)

4) $S_{jk} < K$, $S_{kj} \geq K$ T_j is a parent of T_k.

The third step is the construction of a hierarchy in a form convenient for modification of queries. The authors propose a list structure wherein each term (or concept) owns a list of its parents,a list of its brothers, and a list of its sons. A term with no parents, brothers, or sons is called "isolated"; the phenomenon of isolation is discussed below. If a query is to be generalized, the entries of the parent list of each query term are added to the query vector; if a query is to be specialized, the entries of the sons list of each query term are added; if a query is to be expanded with similar terms, the entries of the brother list of each query term may be added.

These steps are illustrated in the following example.

1) Given the document-term matrix, C:

$$\begin{array}{c|cccc|}
 & T_1 & T_2 & T_3 & T_4 \\
\hline
D_1 & 2 & 0 & 5 & 1 \\
D_2 & 1 & 4 & 1 & 3 \\
D_3 & 4 & 1 & 3 & 0 \\
\hline
\end{array}$$

Derive the term-term matrix, S:

(Steps in calculating S_{12} are illustrated)

$$\sum_{i=1}^{3} C_{i1} = 2 + 1 + 4 = 7$$

$$S_{12} = \frac{1}{7} \cdot [\text{ MIN } (2,0) + \text{ MIN } (1,4) + \text{ MIN } (4,1)]$$

$$= \frac{2}{7}$$

	T_1	T_2	T_3	T_4
T_1	-	2/7	6/7	2/7
T_2	2/5	-	2/5	3/5
T_3	6/9	2/9	-	2/9
T_4	2/4	3/4	2/4	-

2) Choose a cutoff value K and derive relationships:

For $K = 0.50$:

$S_{12} < K$, $S_{21} < K \rightarrow T_1$ and T_2 unrelated

$S_{13} > K$, $S_{31} > K \rightarrow T_1$ and T_3 similar (brothers)

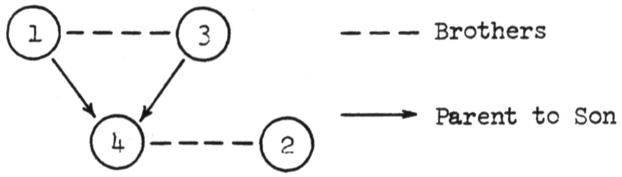
$S_{14} < K$, $S_{41} = K \rightarrow T_1$ parent of T_4

$S_{23} < K$, $S_{32} < K \rightarrow T_2$ and T_3 unrelated

$S_{24} > K$, $S_{42} > K \rightarrow T_2$ and T_4 similar (brothers)

$S_{34} < K$, $S_{43} = K \rightarrow T_3$ parent of T_4

These relationships may be represented graphically:



3) Put entries on the appropriate lists:

Term 1

Parents	(None)
Brothers	3
Sons	4

Term 2

Parents	(None)
Brothers	4
Sons	(None)

Term 3

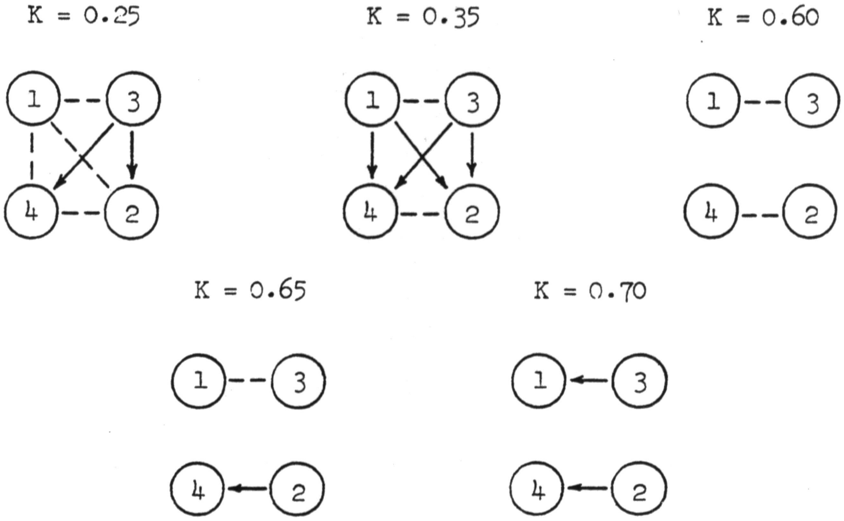
Parents	(None)
Brothers	1
Sons	4

Term 4

Parents	1, 3
Brothers	2
Sons	(None)

A query containing terms 1 and 3 is specialized by adding term 4.

For a given matrix S, varying the cutoff value results in different hierarchies. Referring to the above example, other values of K give the following graphs:



When K = 0, all concepts are brothers. As K increases from zero and reaches the region between S_{jk} and S_{kj}, the brother relationship between T_j and T_k becomes a parent-son relation; as K increases further, these concepts become unrelated. If no entry in the S matric is equal to 1.0, then when K = 1 all concepts are unrelated.

A concept which is unrelated to all other concepts is called "isolated". Two types of isolation may be defined. Consider the entries in a term-term matrix. For a given cutoff value K, a concept is "conditionally isolated" if all entries relating to it are less than K. A concept is "unconditionally isolated" if (1) it is assigned to <u>no</u> document in the collection; <u>or</u> (2) when it is assigned to a document, it is always the <u>only</u> concept assigned. The latter type of concept remains isolated for all $K > 0$.

The above discussion and example illustrate that all information about concept relationships is not contained in one hierarchy constructed for one cutoff value. As K varies from 0 to 1, some relationships endure over a wide range of K-values (say 0.1 to 0.9); these relations are well-defined, or strong. Other relationships appear only once, or over a small range of K-values (say 0.2 to 0.3); these relations are less well-defined, or weak.

While one user may be satisfied with a hierarchy which contains weak relationships, another user may desire a hierarchy containing only very well-defined relationships; neither would be satisfied with a hierarchy which specifies well-defined relationships only in the region of a particular K-value.

The authors suggest a fourth step: the construction of "composite" hierarchies. For a given range R of K-values, a composite hierarchy is generated to include only those relationships which exist over a range $\geq R$. It is possible that brother, parent-son, and "unrelated" relations for a pair of concepts may all exist over ranges $\geq R$; in such a case the authors recommend that the parent-son relation take precedence over the "unrelated" relation. The reason for such a decision rule is that a thesaurus structure, is desired.

This results in a potential benefit to the system user, because composite hierarchies in varying degrees of detail can be made available to him. The degree of detail is a function of the chosen range R and can be characterized by a number (say between 1 and 10). To obtain a particular hierarchy for modifying his queries, the user merely specifies one of the numbers; the corresponding hierarchy is then made available to him.

A continuation of the above example illustrates the construction and the use of composite hierarchies.

First, the relations between pairs of terms are determined as K varies. This may be done in two ways:

a) A set of hierarchies for various values of K between 0 and 1 is constructed, as illustrated by the graphs on page 6.

b) The relations between each pair of concepts are examined as K varies from 0 to 1.

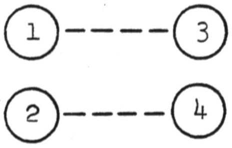
Second, "range tables" are constructed; these display the relationships found and their "duration ranges" (ranges of K-values for which they exist). This is done conveniently if K is varied in uniform increments. The example yields the table shown on the next page; K is incremented by 0.05. For the example the lower bound occurs between 0.20 and 0.25; the upper bound occurs between 0.85 and 0.90; thus K is varied from 0.20 to 0.90.

The range lengths provide a convenient means of assigning numbers to the various composite hierarchies. In this particular example the numbers are merely the range lengths. Composite hierarchies for numbers (range lengths) 9, 5, 4, and 3 are presented.

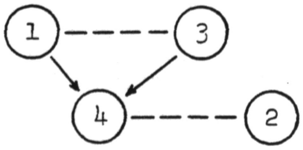
Range Table for the Example

Parent-Son Relations		Brother Relations	
Concept Pair	Range Length	Concept Pair	Range Length
1 → 2	3	1 --- 2	2
3 → 1	4	1 --- 3	10
1 → 4	5	1 --- 4	2
3 → 2	4	2 --- 3	1
2 → 4	3	2 --- 4	9
3 → 4	6	3 --- 4	1

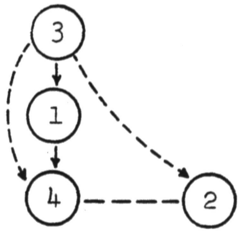
Third, the composite hierarchies are constructed from the range table as follows. The hierarchy for number 9 includes all relations whose range-length values are 9 or greater. Two such relationships exist:



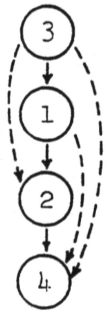
The hierarchy for number 5 includes four relations:



The hierarchy for number 4 includes six relations, two of which overlap. In particular, concept pair (3,1) appears both as a parent-son and as a brother-brother relation. By the precedence rule, however, these concepts are connected by a parent-son relation. The resulting hierarchy is:



In a similar fashion the hierarchy for number 3 is constructed:



The curved, dotted lines denote "grandfather-grandson" relations, which are parent-son relations spanning one or more other levels. Such relations complicate the level structure and obscure ordinary parent-son and brother-brother relations. The authors recommend that grandfather-grandson relations be deleted from each composite hierarchy. For the same reasons, brother-brother relations between concepts on different levels should be deleted.

The composite hierarchies are used as follows. Suppose that the concept numbers represent these concepts:

T_1	library
T_2	dictionary
T_3	information

T_4	thesaurus
T_5	use

Consider the query "THE USE OF A LIBRARY?". The non-trivial words are underlined. This query may be represented by a binary vector of concepts:

1	0	0	0	1

If hierarchy number N = 5 is chosen:

Specialize the query (add sons):

1	0	0	1	1

Then expand the specialized query (add brothers):

1	1	1	1	1

If hierarchy number N = 4 is chosen:

Generalize the original query (add parents):

1	0	1	0	1

If hierarchy number N = 9 is chosen:

Specialize the original query (add sons):

1	0	0	0	1

(No change)

3. Outline of the Investigation

The investigation proceeds in the following stages:

1) Implementation of the program to generate a term-term matrix.
2) Implementation of the program to set up list structures using cutoff values.
3) Implementation of a program to present the list structure and hierarchy in forms convenient for study.
4) Investigation of the effect of varying K for an actual S-matrix. Attempt to confirm theory about variations and range behavior.

Since the aim of this investigation is the study of the techniques and problems involved in automatic generation of hierarchies, and since extensive use of tapes results in processing delays, the programming package is designed for in-core operations. The 100 concepts used are a subset of the 550 concepts in a collection of 82 documents previously used by the SMART system (ADI Collection).

In an actual retrieval system the processing involved in modifying a query uses only the list structure; however, for visual examination of the hierarchy, this structure is not as convenient as a graph. The output program generates a graph similar to those in the examples above. To test the output section of the programming package, a typical hierarchy was constructed containing most of the relationships likely to occur. The output resulting from this example appears in Appendix A.

Using the actual term-term matrix and various cutoff values, the behavior of the hierarchical structure and the range phenomena were studied. The anticipated transitions from brother-brother to parent-son to isolated relations do occur, as illustrated by sample outputs in Appendix A. These outputs are presented in descending order of cutoff value.

Although it is possible to examine the ordinary hierarchies and identify the various ranges for each pair of concepts in the actual term-term matrix, a set of composite hierarchies is not constructed. The authors believe that composite hierarchies should be constructed by examining the relations between each pair of concepts as K varies from 0 to 1 (method b) on page 8); this approach is more direct and requires less time and less memory in the computer. To do this, a composite-hierarchy generating program must be written.

The usefulness of composite hierarchies is best evaluated in actual information-retrieval system. In any event the composite hierarchies must be constructed for the entire set of concepts. Then standard evaluation procedures may be used to compare system performance with composite hierarchies to system performance without them.

References

[1] L. B. Doyle, Is Automatic Classification a Reasonable Application of Statistical Analysis of Text ?, Journal of the ACM, Vol. 12, No. 4, October 1965.

[2] D. Lefkowitz and N. S. Prywes, Automatic Stratification of Information, Proceedings of Spring Joint Computer Conference, Detroit, 1963.

[3] M. Razar and G. Shapiro, Hierarchy Set-up and Hierarchy and Concept-Concept Expansion Procedures, Report No. ISR-9, to National Science Foundation, Section XV, Harvard Computation Laboratory, August 1965.

[4] G. Salton, Manipulation of Trees in Information Retrieval, Communications of the ACM, Vol. 5, No. 2, February 1962.

[5] G. Salton, The Evaluation of Automatic Retrieval Procedures -- Selected Test Results Using the SMART System, American Documentation, Vol. 16, No. 3, July 1965.

[6] G. Salton, Progress in Automatic Information Retrieval, IEEE Spectrum, August 1965.

[7] G. Salton, Information Analysis and Dictionary Construction, Manuscript, Cornell University, 1966.

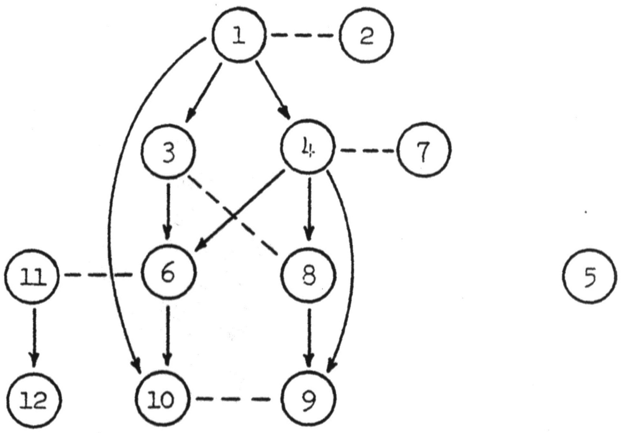
[8] H. F. Stiles, The Association Factor in Information Retrieval, Journal of the ACM, Vol. 8, No. 2, April 1961.

APPENDIX A.

Sample Outputs

Part 1. Hand-Constructed Example

The following structures contain typical relations:



Concept Number	Parents	Brothers	Sons
1	-	2	3,4,10
2	-	1	-
3	1	8	6
4	1	7	6,8,9
5	-	-	-
6	3,4	11	10
7	-	4	-
8	4	3	9
9	4,8	10	-
10	1,6	9	-
11	-	6	12
12	11	-	-

The following printout constitutes output from an actual computer run using the above data.

Output from Hand-Constructed Example

The List Structure from which the Hierarchy is Constructed

Cutoff = (Irrelevant)

```
CONCEPT NUMBER     1

  PARENTS
  BROTHERS     2
  SONS         3    4    10

CONCEPT NUMBER     2

  PARENTS
  BROTHERS     1
  SONS

CONCEPT NUMBER     3

  PARENTS      1
  BROTHERS     8
  SONS         6

CONCEPT NUMBER     4

  PARENTS      1
  BROTHERS     7
  SONS         6    8    9

CONCEPT NUMBER     6

  PARENTS      3    4
  BROTHERS    11
  SONS        10

CONCEPT NUMBER     7

  PARENTS
  BROTHERS     4
  SONS

CONCEPT NUMBER     8

  PARENTS      4
  BROTHERS     3
  SONS         9

CONCEPT NUMBER     9

  PARENTS      4    8
  BROTHERS    10
  SONS

CONCEPT NUMBER    10

  PARENTS      1    6
  BROTHERS     9
  SONS

CONCEPT NUMBER    11

  PARENTS
  BROTHERS     6
  SONS        12

CONCEPT NUMBER    12

  PARENTS     11
  BROTHERS
  SONS
```

All Other Concepts have No Parents, No Brothers, and No Sons.

The Levels and the Concepts on the Levels

Cutoff = 0

```
LEVEL NUMBER    1
    1   BROTHERS    2
        SONS        3     4     10
    2   BROTHERS    1
        SONS
        OVERFLOW COUNT = 0
LEVEL NUMBER    2
    3   BROTHERS    8
        SONS        6
    4   BROTHERS    7
        SONS        6     8     9
    7   BROTHERS    4
        SONS
        OVERFLOW COUNT = 0
```

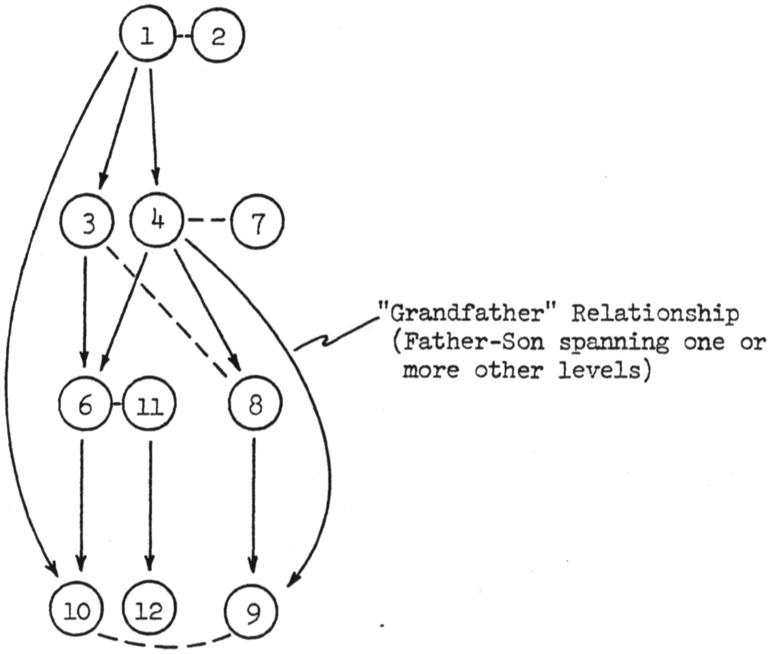
```
LEVEL NUMBER    3
    6   BROTHERS    11
        SONS        10
   11   BROTHERS    6
        SONS        12
    8   BROTHERS    3
        SONS        9
        OVERFLOW COUNT = 0
LEVEL NUMBER    4
   10   BROTHERS    9
        SONS
   12   BROTHERS
        SONS
    9   BROTHERS    10
        SONS
        OVERFLOW COUNT = 0
```

ISOLATED CONCEPTS

```
  5   13   14   15   16   17   18   19   20   21   22   23   24   25   26
 27   28   29   30   31   32   33   34   35   36   37   38   39   40   41
 42   43   44   45   46   47   48   49   50
```

The Hierarchy

Cutoff = 0


1
2
3
4
7
6
11
8
10
12
9
"Grandfather" Relationship (Father-Son spanning one or more other levels)

Part 2. Sample Output Using Actual Data

The following are samples of output from a run using the actual term-term matrix. These samples are presented in order of decreasing cutoff value. The first example includes the list structure for cutoff value K = 0.20, the level structure for K = 0.20, and the hierarchy graph for K = 0.20; for the other values of K, only the graphs are shown.

As an illustration of the transition phenomenon, consider the relation between concepts 75 and 85. When K = 0.20, these concepts are isolated; when K = 0.18, 85 is the parent of 75; when K = 0.085, they are brothers.

The List Structure from which the Hierarchy is Constructed

Cutoff = .200000

```
CONCEPT NUMBER    1                                 CONCEPT NUMBER   69
 PARENTS                                             PARENTS
BROTHERS                                            BROTHERS   50
SONS        60                                      SONS

CONCEPT NUMBER   50                                 CONCEPT NUMBER   91
 PARENTS                                             PARENTS
BROTHERS   69                                       BROTHERS 100
SONS                                                SONS

CONCEPT NUMBER   60                                 CONCEPT NUMBER  100
 PARENTS     1                                       PARENTS
BROTHERS                                            BROTHERS   91
SONS                                                SONS
```

All Other Concepts have No Parents, No Brothers, and No Sons.

List Structure, Continued -- Isolated Concepts

```
 2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
32  33  34  35  36  37  38  39  40  41  42  43  44  45  46
47  48  49  51  52  53  54  55  56  57  58  59  61  62  63
64  65  66  67  68  70  71  72  73  74  75  76  77  78  79
80  81  82  83  84  85  86  87  88  89  90  92  93  94  95
96  97  98  99
```

The Levels and the Concepts on the Levels

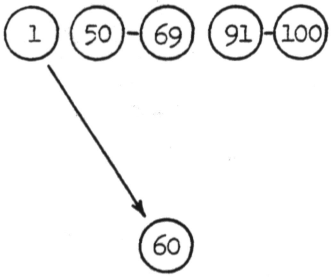
Cutoff = .200000

```
LEVEL NUMBER    1
     1    BROTHERS
          SONS       60
    50    BROTHERS   69
          SONS
    69    BROTHERS   50
          SONS
    91    BROTHERS  100
          SONS
   100    BROTHERS   91
          SONS
          OVERFLOW COUNT = 0
```

```
LEVEL NUMBER    2
    60    BROTHERS
          SONS
          OVERFLOW COUNT = 0
```

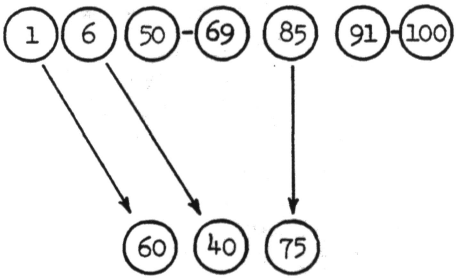
The Hierarchy

Cutoff = .200000



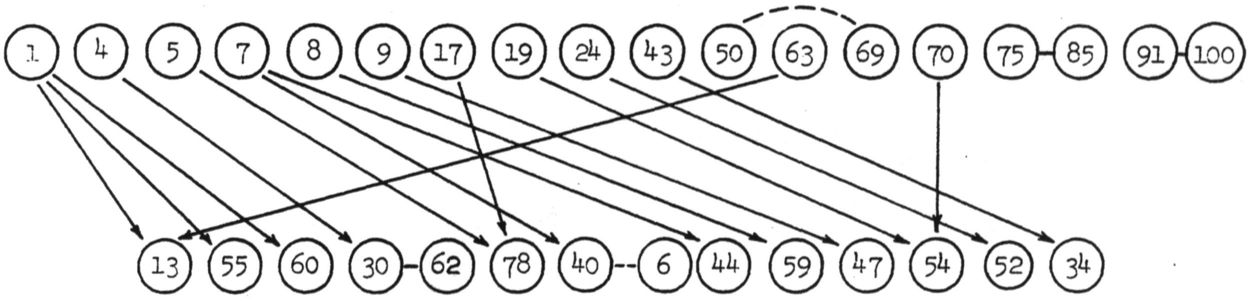
The Hierarchy

Cutoff = .180000



The Hierarchy

Cutoff = .085000



82 Document A.D.I. Collection Thesaurus

1 INFORMATION
2 SUBSYSTEM
SYSTEM
3 COMPUTER-BASED
COMPUTER
4 INDEX
5 INFORMATION-RETRIEVAL
IR
RECALL
RECOVER
RETRIEVE
6 TECHN
7 PROCESS
8 DOCUMENT
9 LIBRAR
LIBRARY-SIZED
10 SCIENCE
SCIENTIFIC
11 CAREER
DESIGNER
DOCUMENTALIST
ENGINEER
EXPERT
INVESTIGATOR
PHYSICIST
PRACTITIONER
PROFESSION-ORIENTED
PROFESSIONORIENTED
PROFESSION
SCIENTIST
SPECIALIST
SUB-PROFESSIONAL
12 BROWSE
CONSULT
LOOK-UP
LOOK
LOOKUP
PERUSE
SEARCH
13 BORROWER
CLIENTELE
CLIENT
CONSUMER
CUSTOMER
ENQUIRER
INQUIRER
INVESTIGATOR
PATRON
QUESTIONNER
READER
RECIPIENT
REQUESTER
REQUESTOR
RESEARCHER
SEARCHER
USER-ORIENTED
USER
14 ADVICE
ADVISE
SERVICE
15 EMPIRICAL
EXPERIMENT
PROGRAM
16 ADDER
ALGORITHM
BUFFER
COMPILER
PREPROCESS
PROGRAMED
PROGRAMER
PROGRAMING
PROGRAMMED
PROGRAMMER
PROGRAMMING
REAL-TIME
ROUTINE
17 DATA
FACTS
18 CENTER
CENTRAL
CENTRE
CLEARINGHOUSE
CLEARING-HOUSE
DECENTRALIZE
FOCUS
HEADQUARTER
SEMICENTRAL
19 AUTOMATE
MACHINE
MECHAN
20 DICTIONARY
LEXIC
THESAURI
THESAURUS
VOCABULARY

21 ARTICLE
BULLETIN
ISSUE
JOURNAL
LETTER
MAGAZINE
NEWSPAPER
PERIODICAL

22 ARRANGE
DECIDE
DECISION
ORGANIZATION
ORGANIZE
PLAN
POLICY
PROJECTION
STRATEGY

23 ASSOCIATION
BOARD
FACILITY
FEDERATION
FOUNDATION
INSTALLATION
INSTITUTE
ORGANIZATION
SOCIETY
UNIVERSITY

24 CARBON
DUPLICATE
FACSIMILE
PHOTOCOPY
REPLICA
REPRODUCE
REPRODUCT
REPRO
TRANSCRIBE
TRANSCRIPT

25 FILE

26 COMMUNIC
MESSAGE

27 BUSHIPS

28 READ

29 PERMUTE

30 ROLE

31 CAMPUS
COLLEGE
GRADUATE
SCHOOL
STUDENT

32 MATERIAL
TEXT
TEXTUAL

33 ANAL

34 ACADEMIC
CANDIDATE
CORE
COURSES
CREDIT
CURRICUL
DEGREE
DOCTOR
EDUC
ELECTIVE
ENROLL
EXAMINE
FACULTY
INSTRUCT
LECTURE
MASTER-S
NON-CREDIT
TRAIN

35 COLUMN
DOUBLE-COLUMN
DOUBLE-SPACE
FORMAT

36 CROSS-REFERENCE
CROSSREFERENCE
CROSS
PERIPHERAL
REFER

37 CHOICE
CHOOSE
CHOSEN
CHOSE
ELECTIVE
ELECT
LOCATE
PREFER
SELECT

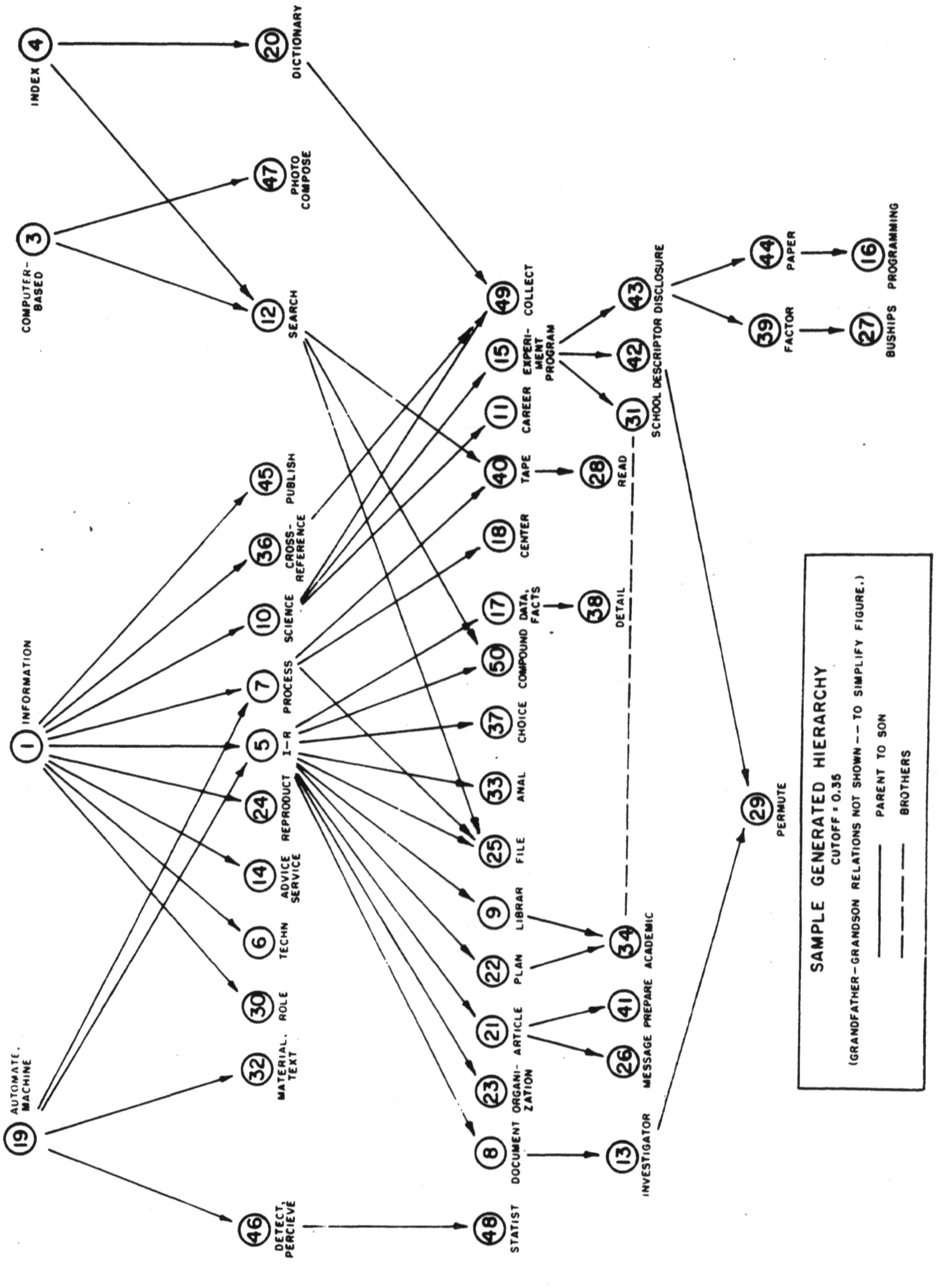
38 DETAIL
DISTINCT
NONCONVENTIONAL
NON-CONVENTIONAL
REFINE
SPECIAL
SPECIFIC

39 FACTOR
SPECIFICATIONS
STANDARDS

40 STRIP
TAPE

41 PREPARE

42 DESCRIPTOR
INDICATOR
KEYWORD
UNITERM
43 DISCLOSURE
DISSERTATION
DRAFT
MANUALS
MANUSCRIPT
MONOGRAPH
NEWSLETTER
NEWS
PAPERS
PATENT
REPORT
REPRINT
44 PAPER
45 PUBLICATION
PUBLISH
46 DETECT
DISCRIMIN
DISTINGUISH
PERCEIVE
PERCEPT
RECOGN
47 LINOFILM
PHOTOCOMPOSE
PHOTOCOMPOSITION
PHOTO-COMPOSE
PHOTO-COMPOSITION
PHOTO-OFFSET
PHOTO-PRINTER
PHOTOLITHOGRAPH
PHOTOTYPESET
48 BIOSTATIST
STATIST
49 COLLECT
COMPENDI
COMPILE
50 CARBON
CATAL
CHEM
COMPOUND
MOLECULE
ORGANIC



Sample Hierarchy