DOCUMENT RESUME

ED 072 826 LI 004 180

AUTHOR	Lay, William Michael
TITLE	The Double-KWIC Coordinate Indexing Technique: Theory, Design, and Implementation.
INSTITUTION	Ohio State Univ., Columbus. Computer and Information Science Research Center.
SPONS AGENCY	National Science Foundation, Washington, D.C. Office of Science Information Services.
REPORT NO	OSU-CISRC-TR-73-1
PUB DATE	Feb 73
NOTE	263p.;(0 References); Dissertation
EDRS PRICE	MF-$0.65 HC-$9.87
DESCRIPTORS	*Automatic Indexing; *Coordinate Indexes; *Indexes (Locaters); *Indexing; *Information Retrieval; Relevance (Information Retrieval)
IDENTIFIERS	DKWIC; Double KWIC Coordinate Index; *Key Word in Context; KWIC

ABSTRACT

The development of an automatic indexing technique, called Double KWIC (DKWIC) Coordinate Indexing, is described which extends the KWIC indexing principles to provide easy access to an additional level of specificity for information indexed under these frequently appearing terms. Chapter 2 discusses indexing terminology and some fundamental relationships between indexing and document retrieval. Chapter 3 sketches a brief history of automated indexes describing frequently encountered methods of construction and display. Chapter 4 introduces the Double-KWIC Coordinate Indexing scheme and discusses its advantages and disadvantages relative to several other KWIC indexing schemes. Chapter 5 discusses refinements in the prototype indexing scheme which led to the production of KWOC-DKWIC hybrid indexes. Chapter 6 considers the problems of vocabulary control in a natural language environment. Several methods of automated vocabulary normalization are described. Chapter 7 examines the role played by the index analyst in creating a Double-KWIC Coordinate Index and resolves the plaguing problem of main term selection by an automatic selection algorithm which can only be applied successfully with KWIC-DKWIC hybrid indexes. The final chapter examines the parametric controls of the KWIC-DKWIC indexing scheme and discusses some relationships among these parameters and the indexes produced. (Author/NH)

(OSU-CISRC-TR-73-1)

THE DOUBLE-KWIC COORDINATE INDEXING TECHNIQUE: THEORY, DESIGN, AND IMPLEMENTATION

by

William Michael Lay

Work performed under

Grant No. 534.1, National Science Foundation

Computer and Information Science Research Center
The Ohio State University
Columbus, Ohio 43210

February 1973

PREFACE

This work was done in partial fulfillment of the requirements for a doctor of philosophy degree in Computer and Information Science from The Ohio State University. It was supported in part by Grant No. GN 534.1 from the Office of Science Information Service, National Science Foundation, to the Computer and Information Science Research Center of The Ohio State University.

The Computer and Information Science Research Center of The Ohio State University is an interdisciplinary research organization which consists of the staff, graduate students, and faculty of many University departments and laboratories. This report is based on research accomplished in cooperation with the Department of Computer and Information Science.

The research was administered and monitored by The Ohio State University Research Foundation.

ACKNOWLEDGMENTS

I would like to express my appreciation to the many people who contributed to the successful completion of this work.

I am indebted to Professor Anthony Petrarca, my advisor, who initiated this investigation and whose valuable assistance and occasional prodding immeasurably aided the progress and fruition of this work. I am very grateful to the Professors James Rush and Lee White for serving as members of the committee who read this dissertation.

I am appreciative of Professor William Atchison who allowed my continuance of this work while I was teaching at the University of Maryland and to Mr. Robert Jones of the Health Sciences Computer Center of the University of Maryland who allowed me to use the HSCC computing facilities to test some of the programs designed and to produce this document.

Partial support of this work has been provided by a grant (GN-534.1) from the National Science Foundation to the Computer and Information Science Research Center, by the Ohio State University Instruction and Research Computer Center who donated much of the computer time, and through a Title II-b Fellowship in Library and Information Science awarded by the Office of Education.

Finally, I would like to express my gratitude to my wife, Carolyn, who endured the years I spent as a graduate student lending hardy moral and sometimes physical support to this work.

TABLE OF CONTENTS

LIST OF FIGURES

Page

Page

LIST OF TABLES

CHAPTER I. INTRODUCTION: THE NEED FOR BETTER INDEXING PRACTICE

> "...unless this mass (of information) be properly arranged and the means furnished by which its contents may be ascertained, literature and science will be overwhelmed by their own unwieldy bulk."
>
> Annual Report of the Smithsonian Institute for 1851

For more than a century this warning given explicitly by John Henry, Director of the Smithsonian Institute in 1851, went unheeded. He forsaw a potential unsurmountable barrier of literature when the total increment to man's published works was estimated at 20,000 volumes annually. Henry's statement was ignored as were others issued from time to time by those who saw the impending danger buried beneath the accumulating bulk of literature.

The inevitable explosion accompanied by a frantic call for control came during the boom following World War II. The world's research effort, stimulated by a war-time environment, produced a new flood of literature so great that the existing methods of information dissemination could no longer be considered adequate. Simultaneously, such a realization was evolving within the scientific community. Research could be increasingly stimulated by an intelligent insight into what had gone before or what had been reported in the literature. It was ironic that the recognition of

the failure of traditional dissemination techniques should accompany man's greatest need for information control!

Not until that time did man finally acknowledge that the traditional library tools were not only inadequate but actually limiting his ability to cope with the many new problems that faced him. He required highly specialized information currently being spawned by the scientific community as well as those past explorations buried deep beneath the "unwieldy bulk." He was thwarted by the necessary time lag of traditional techniques and severely restricted by the conventional indexing schemes. He was frustrated by:

a) the physical impossibility of his reading and remembering all of the literature that could have a reasonable probability of being of interest at some unspecified future time;

b) the economic impossibility that he could process a major part of the literature for later exploitation that exhibited probable interest;

c) the mechanical impossibility that the currently employed literary procedures could effectively cope with his highly specialized requests.

Dr. Vannevar Bush in a report to the President and later in an often quoted paper (Bush,45) focused attention on a most critical deficiency in traditional library practices:

> "...The difficulty seems to be, not so much that we publish unduly in view of the extent and variety of present day interests, but rather the publication has been extended far beyond our present ability to make real use of the record. The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentary important item is the same as was used in the days of square-rigged ships... The real heart of the matter of selection, however, goes deeper than a lag in the adoption of mechanisms by libraries, or a lack of development of devices for their use. Our ineptitude in getting at the record is largely caused by the artificiality of the systems of indexing..."

The overwhelming need for literature retrieval combined with Bush's observations on traditional indexing methods prompted many researchers to directly attack the problems frustrating the library users. The advent of electronic machines used to manipulate non-numeric data spurred the development of mechanized approaches to indexing and library management.

Considering Bush's comments, the study of these types of problems should more aptly be entitled "information storage <u>for</u> retrieval." The literature still abounds with data, conclusions, opinions, and theories applicable to a host of fields and recorded in journals, reports, proceedings, and theses too numerous to comprehensively list.

The need for high-quality printed indexes has not diminished despite the recent strides in automatic information retrieval systems. Since the application of

key-word-in-context (KWIC) indexing (and key-word-out-of-context (KWOC) indexing) as an automated derivative indexing technique {Luhn,59}, the KWIC index has been used widely but not without some dissatisfaction with its quality as a retrieval tool {Fischer,66}. Most attempts to improve its quality have dealt with variations in format to improve readability, or with enrichment terms to provide additional index entries which otherwise would not have been derived from the words in the titles. Neither of these modifications improve the quality of the index when an index term appears frequently in the title phrases indexed. In this case, index terms form large blocks of index entries where access to more specific concepts is hindered by the random scattering of secondary concepts in each index phrase. The user must scan the context about each term in the block in order to determine that subset of entries which is pertinent to a more specific search.

This thesis describes the development of an automatic indexing technique, called Double-KWIC (DKWIC) Coordinate Indexing, which extends the KWIC indexing principles to provide easy access to an additional level of specificity for information indexed under these frequently appearing terms. Chapter 2 discusses indexing terminology and some fundamental relationships between indexing and document retrieval important to the chapters that follow. Chapter 3 sketches a brief history of automated indexes describing

frequently encountered methods of construction and display. Chapter 4 introduces the Double-KWIC Coordinate Indexing scheme and discusses its advantages and disadvantages relative to several other indexing schemes based on KWIC indexing principles. Chapter 5 discusses refinements in the prototype indexing scheme which led to the production of KWOC-DKWIC hybrid indexes. Chapter 6 considers the problems of vocabulary control in a natural language environment. Several methods of automated vocabulary normalization are described which provide a basis for an effective automated solution to some scattering problems in printed indexes. Chapter 7 examines the role played by the index analyst in creating a Double-KWIC Coordinate Index and resolves the plaguing problem of main term selection by an automatic selection algorithm which can only be applied successfully with KWIC-DKWIC hybrid indexes. The final chapter examines the parametric controls of the KWIC-DKWIC indexing scheme and discusses some relationships among these parameters and the indexes produced. Some concluding remarks spell out areas where this indexing method can be modified further to supply even more useful indexes. Appendix C of this thesis acts as a documentation guide to the computer programs written to generate KWOC-DKWIC and KWIC-DKWIC indexes, with or without vocabulary control. A KWIC-DKWIC index of this document prepared from the phrases appearing in the Table of Contents, List of Tables, and List of Figures serves not

only as an example of the indexing system described in this thesis but also provides an index to important topics of the thesis.

CHAPTER II. INDEXING TERMINOLOGY AND SOME FUNDAMENTAL RELATICNSHIPS BETWEEN INDEXING AND DOCUMENT RETRIEVAL

Since this thesis deals with the automatic construction of useful indexes to collections of documents, a few definitions and relationships appropriate to the general topics of indexing and document retrieval are presented in this chapter. A document is an identifiable collection of concepts which can be considered as a single unit. A journal or journal article, a chapter of a book, a paragraph of a chapter, or an entire book can be considered as a document. A document may be something other than conventional printed matter, such as a file recorded on magnetic tape or a motion picture film. In general, a document will assume three attributes: a title, a body, and an accession code. A title is a condensed description of the contents of the document body and usually consists of several phrases composed of high-content words. The body of a document contains a discussion of the relationships existing among the concepts described therein while an accession code is a coded identifier of the document.

An index is a document consisting of an ordered set of index entries. Each index entry describes, via an index term, a subset of the concepts found in an identifiable class of documents and contains a means of locating this

class of documents. For example, an index commonly found in the back of most books consists of index entries listed alphabetically (an ordering) on the basis of the important topics (index terms) discussed in the text. The documents in which these concepts are described are identified and located by page number (accession code). Here, a document is equivalent to a page and the class of documents identified by the index entry consists of a list of page numbers. A single index entry rarely provides information concerning every concept described in the document it identifies, as the example above implies. Consequently, the topic discussed on the pages noted in an index entry may be one of many discussed within the body of the indicated page. In this example, it was assumed that the page numbers listed in the index entry referred to pages of the text containing the index. This may seem to be a trivial point, but its importance becomes more apparent when large collections of documents are to be indexed.

The means of locating a document, its accession code, may be much broader in scope to aid the retriever. For example, in Chemical Titles and other publications produced by Chemical Abstracts Service (CAS,72), documents (journal articles) are identified by a 17 character field which includes a coded journal title (ASTM coden), its volume and page number. Libraries employ an accession coding scheme which reflects the subject matter of the document as well as

its shelf location within the library (see Dewey,65). Regardless of its length or usefulness to the retriever, the accession codes assigned to documents of a collection will be assumed unique.

It is sometimes convenient to view an index as a mapping of a document space, D, into an ordered index space, I.

f:D -> I

The indexing function, f, relates elements of D, documents, to corresponding elements of I, index entries.

For every document, d, in D, there exists a set of index descriptors generated by applying the indexing function to the document. Thus,

set of index descriptors of d<j>
= f(d<j>) = {i<1>,i<2>,...,i<n<j>>}<j> *

That is, for each document of D there exists a set of index descriptors in I which describe the concepts contained in the document. The number of index entries generated from the above descriptors, n, is a measure of the identified (and accessible) concepts of the document d, and is sometimes referred to as the breadth of indexing. The depth of indexing refers to the amount of detail about the concept

---

* The notation used in the above equation and elsewhere in this thesis deviates slightly from the notation normally used because of the limited character set available for keyboarding of this thesis which was processed and printed by computer text processing programs. The form of the notation used for this thesis is summarized in the Glossary.

described by an index entry. The application of the indexing function to a document producing a set of index descriptors is called <u>indexing</u> .

Similarly, there exists a type of inverse function, g, which maps the index space into the document space.

$$g: I \rightarrow D$$

For each entry in I, there exists a set of <u>document descriptors</u> generated by the function, g.

$$\text{set of document descriptors of } i_k = g(i_k) = \{d_1, d_2, \ldots d_{m_k}\}_k$$

Therefore, the function, g, relates a subset of the documents in D having a common concept represented by the index entry, i. The cardinality of the document descriptor, m, indicates the number of documents located by the mapping function, g. The function, g, describes the action of document retrieval by the generation of document descriptors. Consequently, g will be referred to as the <u>retrieving</u> function.

Before a more thorough analysis of the functional characteristics of indexing and retrieving are examined, let us characterize some of the properties of the sets of documents and index entries.

When the elements of the index are just single words or short descriptive phrases accompanying the accession code, then the index is related to a <u>uniterm</u> index as developed by Taube {Taube,61}. If these single terms can be reduced in

scope by the application of one or more levels of subterms, then the index is called a coordinate index after Johnson (Johnson,59).

If the index entries describing document concepts are condensed into words or phrases possibly not found in the document itself but considered to be likely and useful index terms, then the function of indexing is called assigned. The term derivative indexing is used to describe the indexing function when the index entries are extracted from the title or body of the document.

Many indexes are restricted to a fixed vocabulary. The index terms forming the set I are predetermined, requiring that the indexing function, f, always generate index descriptors within this set for each new document added to the collection. Consequently, assigned indexing techniques are generally required for fixed vocabulary indexes. In this restrictive sense, a fixed vocabulary index is usually accompanied by an authority list which directs the retrieving function to a preferred index entry for other concepts not found in the index itself. The authority list may be included in the index space itself in the form of "see" cross references which list the corresponding preferred index entry as an indirect reference.

When a free vocabulary is used to create index descriptors for documents entering the collection, each application of the indexing function is independent of any

other indexing operation. The addition of documents to a collection can cause an increase in the number of index terms found in the index. Derivative indexes commonly use this technique. As a result of the freedom reflected in the indexing function and the redundancy of natural language, a particular concept may appear in many places in the index. Even the same word used to describe a concept may appear in various inflectional forms. A useful restriction of the vocabulary freedom replaces inflectional variations of words with a common preferred form.

Let us now turn our attention to the indexing and retrieving functions. Some useful results can be gleaned from their functional relationships if first a null operation is defined.

Let PHI<I> and PHI<D> represent the null index entry and document respectively. Define

$$f(PHI\langle D\rangle) = PHI\langle I\rangle$$
$$g(PHI\langle I\rangle) = PHI\langle D\rangle$$

Then the operations of union and intersection can be defined. (The operations will be carried out using the retrieving function only; however, the results hold for the indexing function as well.)

$$g(i\langle k\rangle \text{ UNION } i\langle j\rangle) = g(i\langle k\rangle) \text{ UNION } g(i\langle j\rangle)$$

$$g(i\langle k\rangle \text{ INTERSECT } i\langle j\rangle) = \begin{cases} PHI\langle D\rangle \text{ for } k \neq j \\ g(i\langle k\rangle) \text{ for } k = j \end{cases}$$

Since the index entries are assumed unique, the operation of intersection is non-null within the index space only when the index entries are identical.

These two operations lead to the foundations of document retrieval through the retrieving function. If X and Y are subsets of index entries, then

$$g(X \text{ UNION } Y) = g(X) \text{ UNION } g(Y)$$
$$g(X \text{ INTERSECT } Y) \leq g(X) \text{ INTERSECT } g(Y)$$
where X, Y are contained in I

The document descriptor formed by the union of two sets of index terms follows trivially. However, intersection in the index space is not equivalent to intersection in the document space. Without loss of generality, let us assume that the elements of X and Y can be separated into three distinct subsets, A, B, and C such that

$$X = A \text{ UNION } B$$
$$Y = A \text{ UNION } C$$
$$B \text{ INTERSECT } C = A \text{ INTERSECT } B = A \text{ INTERSECT } C = PHI\langle I\rangle$$

then,

$$g(X \text{ INTERSECT } Y)$$
$$= g((A \text{ UNION } B) \text{ INTERSECT } (A \text{ UNION } C))$$
$$= g(A \text{ UNION } (B \text{ INTERSECT } A) \text{ UNION } (B \text{ INTERSECT } C) \text{ UNION } (A \text{ INTERSECT } C))$$
$$= g(A) \text{ UNION } PHI\langle D\rangle \text{ UNION } PHI\langle D\rangle \text{ UNION } PHI\langle D\rangle$$
$$= g(A)$$

however,

$$g(X) \text{ INTERSECT } g(Y)$$
$$= g(A \text{ UNION } B) \text{ INTERSECT } g(A \text{ UNION } C)$$
$$= (g(A) \text{ UNION } g(B)) \text{ INTERSECT } (g(A) \text{ UNION } g(C))$$
$$= g(A) \text{ UNION } (g(B) \text{ INTERSECT } g(A)) \text{ UNION } (g(B) \text{ INTERSECT } g(C)) \text{ UNION } (g(A) \text{ INTERSECT } g(C))$$

but since $g(A) \geq g(B) \text{ INTERSECT } g(A)$ and $g(A) \geq g(A) \text{ INTERSECT } g(C)$

Then g(X) INTERSECT g(Y)
= g(A) UNION g(B) INTERSECT g(C)
and g(B) INTERSECT g(C) may be non-null
Consequently
g(X) INTERSECT g(Y)
= g(X INTERSECT Y) UNION g(B) INTERSECT g(C)
= g(X INTERSECT Y) UNION
g(X INTERSECT ¬Y) INTERSECT
g(¬X INTERSECT Y)

The relationships above depict the common actions performed by a retriever using an index. The union of index entries retrieves documents containing any of the concepts described by the entries. Because of the uniqueness of index entries, the intersection of concepts is carried out in the document space instead of the index space. When the subsets X and Y are mutually exclusive, as is the usual case, the desired retrieval can only be performed in the document space.

When an index has been adequately prepared, the retrieval function is represented by a mechanical procedure of tracing the location of the documents via the accession codes contained in the index entry. The performance of an index to accurately retrieve pertinent documents is not a reflection of the mechanical retrieving function but a consequence of a poorly constructed index descriptor by the indexing function.

Real indexing functions suffer from two general types of errors:

1) attribute only a subset of the concepts found in a document to the document;

2) attribute to a document a set of concepts not present in the document.

These errors may be examined formally by introducing a perfect indexing function, f'. Let

f(d) = A UNICN B for all d in D

where

A = {index entries describing concepts in d attributed to d}

B = {index entries describing concepts not in d attributed to d}

The perfect indexing function, f', would generate an index descriptor of the form:

f'(d) = A UNICN C for all d in D

where A is defined above

C = {index entries describing concepts in d not attributed to d by f}

Let |X| represent the number of elements in the set X. Then, the real index generated by applying f to the entire document collection can be represented as:

I = (i=1,|D|) UNION f(d<i>)
= (i=1,|D|) UNION (A<i> UNION B<i>)

Should any intersections of the sets A and B be non-empty, irrelevant documents will be retrieved when the retrieving function is applied to any member of that set. That is, if

g(A<i> INTERSECT B<j>) ≠ PHI<D>
for some i, j in {1,2,...,|D|}

then some irrelevant documents will be retrieved <u>regardless of the perfection of g.</u> If

(i=1,|D|) UNION B<i> is contained in
(i=1,|D|) UNION A<i>

then every retrieval will at least produce one relevant document. The only method of decreasing the number of irrelevant documents retrieved lies in reducing the set B of improperly attributed document concepts - a refinement of the indexing function.

Applying the perfect indexing function to the document collection, a superset of the real index is built:

I is contained in
(i=1,|D|) UNION f'(d<i>)
= (i=1,|D|) UNION (A<i> UNION C<i>)

A non-empty intersection of the sets A and C leads to the possibility of not retrieving all the relevant documents pertaining to a concept described by an index entry. Consequently, if

g(A<i> INTERSECT C<i>) ≠ PHI<D>
for some i, j in {1,2,...,|D|}

then a retrieval error occurs regardless of the perfection of g. This type of error is masked from the user since it reflects relevant documents not retrieved.

These abstract set notations can be transformed to the more familiar measures of retrieval effectiveness of recall and precision.

$$\text{Recall} = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents in data base}}$$

$$\text{Precision} = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

Let X in I represent a function of index terms.

Let $x = X \text{ INTERSECT } A$
$y = X \text{ INTERSECT } B$
$z = X \text{ INTERSECT } C$

Then the documents retrieved from the real index are

$$g(X) = g(x \text{ UNION } y) = g(x) \text{ UNION } g(y)$$

while those retrieved from the ideal index are

$$g(X) = g(x \text{ UNION } z) = g(x) \text{ UNION } g(z)$$

then

$$\text{Recall} = |g(x)|/|g(x) \text{ UNION } g(z)|$$
$$= |g(x)|/(|g(x)|+|g(z)|-|g(x) \text{ INTERSECT } g(z)|)$$

$$\text{Precision} = |g(x)|/|g(x) \text{ UNION } g(y)|$$
$$= |g(x)|/(|g(x)|+|g(y)|-|g(x) \text{ INTERSECT } g(y)|)$$

Note that recall and precision are inversely related to the inaccuracy of the indexing function.

The reader should be convinced by these last arguments that the failures found in real document retrieval systems are not in the retrieval network per se. This can be reduced to a mechanical procedure of performing transformations on accession codes. At best, the retrieval network performs in a fashion proportional to the perfection of the index on which it is based. Consequently, the goal of this thesis is to provide an automatic indexing technique to produce higher quality indexes.

CHAPTER III. AUTOMATED INDEXING: A BRIEF HISTORY

The application of derivative techniques to documents predates electronic machines by centuries. Several orders of monks during the 12th and 13th centuries manually prepared concordances (Simmons,63), listings of each word with all the contexts in which it appeared in a document. Concordance construction is an index producing operation, an indexing function that preserves the contents of the full document. However, such exhaustive concordances are incredibly time consuming, tedious, and error prone tasks when carried out manually. A suggestion as early as 1856 was proposed to use concordance techniques to generate an index from titles of document collections (Simmons,63), but the necessary manual preparation time caused the idea to be dropped.

The advent of general purpose electronic computers promised non-numeric processes which could represent, preserve, manipulate, and print textual data at unprecedented speeds. Because the computer could faithfully reproduce the textual transformations, most of the previous deficiencies and clerical labor of the manual production of concordance-like indexes could be reduced to preparing a corpus of documents in machine readable form. Even more radical possibilities for the potential use of computers was

envisioned by many of the pioneers of the time. The salient features of some of these systems of indexing will be discussed in this chapter. These methods can be generally classified by the processing operations automatically applied to the text of a document. A computer-compiled index is merely an ordering of permutations of preselected items (index entries) presented for input. The index terms of even the most elementary form of a computer-generated index have been extracted from the input text by some automated selective procedure. In either case, the ordering and duplicating of index terms, the compilation and presentation of accession codes, and the formatting and printing of the index are computer controlled. The amount of intellectual effort required to augment the automatic process is an attribute of the particular system and is not amenable to general classification.

3.1. Computer-Compiled Indexes

One of the first and obvious applications of computers to index construction was the manipulation of index entries previously selected by human analysis. The power of a computerized technique of duplicating and sorting index entries could provide various orderings and listings of terms for special purpose indexes. For example, from the same machine readable data base, a uniterm index could be prepared as well as an author index. These by-products of machine-readable indexes were recognized as being as

important as the index itself (Olney,63). Not only could duplicate copies of the index as a whole be prepared, but the basis for elementary automated retrieval systems was also present. From a single machine-readable uniterm index, a specified subset of the index entries could be listed as a special purpose index, or, with a slightly more sophisticated program, listings of documents having more than one common index entry could be prepared.

Completely new types of indexes, previously considered unmanageable because of the required tedious manual labor, could be considered. Recall that the indexing function maps documents into index descriptors. When a uniterm index is constructed, each entry is a subject heading (uniterm) consisting of a single keyword, or several keywords (or a code representing these keywords) and the document accession code. A new index term can be constructed from the concatenation of the terms of the index descriptor. That is, if

$$f(d) = \{i<1>,i<2>,...i<n>\}$$

where i<j> represents a uniterm, then the new term i' is

$$i' = i<1>i<2>...i<n>$$

This new term provides much more information to the user since all the descriptors ascribed to the document are present. Indeed, the depth of the index term is increased, but, if this were the only entry under which the document may be found, and the ordering of the index is alphabetical

by entry, a user will be led to document d only through the term i', a definite decrease in breadth. For example, titles of new books produced by some publishing houses (see McGraw-Hill,72) are ordered in lists by the first word of the title. A title found in these lists closely models an index term consisting of the concatenation of descriptors when each significant title word is considered to be a descriptor. A solution to the problem of accession to only the first word of the list would be to construct a rotated keyword index, discussed in the next section.

3.1.1. Rotated Keyword Index

In a rotated keyword index, an index term is constructed beginning with each uniterm followed by the remaining uniterms assigned to the document as if the terms were formed by successive uniterm rotations. For example, if

$$f(d) = \{a,b,c\}$$

then

$i'\langle 1\rangle$ = abc
$i'\langle 2\rangle$ = bca
$i'\langle 3\rangle$ = cab

Rotated keyword indexes retain the same breadth while increasing the depth of uniterm indexes. Skolnik has demonstrated the usefulness of a rotated keyword index which he calls the MULTITERM index (Skolnik,70). When the entries are ordered alphabetically, documents having at least one uniterm in common are listed together. If two documents

share more than one uniterm they may be separated in the index by an arbitrary number of unrelated entries which depends upon the order in which the uniterms were concatenated to form the initial index entry. The random distribution of index entries sharing more than one uniterm reduces the effectiveness of rotated keyword indexes for performing coordinate searches.

3.1.2. Completely Permuted Keyword Index

All index terms having an arbitrary number of uniterms in common are collected in a single place in a completely permuted keyword index . Instead of forming the cyclic rotations of the uniterms, the indexing function produces all permutations of the uniterms as index entries.

Thus, if

$$f(d) = \{a,b,c\} .$$

then

$$i'\langle 1\rangle = abc$$
$$i'\langle 2\rangle = acb$$
$$i'\langle 3\rangle = bac$$
$$i'\langle 4\rangle = bca$$
$$i'\langle 5\rangle = cab$$
$$i'\langle 6\rangle = cba$$

Coordinate searches require only one entrance into the index beginning with the entry associated with any combination of uniterms of interest.

Completely permuted keyword indexes suffer a size problem and, because of this, no concrete example can be cited. If an indexing function produces, on the average, n uniterms per document, then a rotated keyword index contains

m * n index entries (m = number of documents in the collection) while a completely permuted keyword index would contain m * n! entries. Ten keywords is not an uncommon number to be assigned to a document. For a collection of one hundred thousand documents, 1,000,000 entries would be included in a rotated keyword index, but each document assigned 10 entries would be entered 3,628,800 times in a completely permuted keyword index! Although a computer may not be disturbed by the size of such an index, the user may (as would the producer paying for its creation). Consequently, other means for achieving coordinate searches were considered.

3.1.3. Selected Listing In Combination (SLIC) Index

Undoubtedly, a completely permuted index provides for document retrieval through any ordering of terms assigned to a document, but as Sharp [Sharp,66] has pointed out, "this multiplicity of entries is not only extravagart but quite unnecessary." The requirement of a coordinating system is to provide the searcher with all combinations of terms pertinent to both the searcher and document concerned. All combinations (in the mathematical sense) of index terms together with a canonical scheme for representing them suffice as useful coordinate entries for indexing.

To consider the indexing function, let n be the number of uniterms assigned to a document. The index should include every combination from 1 to n, every combination

from 2 to n, ... , and every combination from n to n of assigned terms. The size problem found in a completely permuted index is considerably reduced since the total number of terms can be expressed as

(i=1,n) SUM (c<n,i>)
= 2**n - 1 (note for n>3 this is less than n!)

Each combination of terms generated must be unique for the retrieval function to operate successfully; consequently, some ordering relationship must be applied to each combination. The obvious order for an index using natural language terms is alphabetical. Assuming an indexer has assigned the terms a,b,c, and d to a document and a canonic alphabetical ordering is observed, then the index terms generated follows:

1 a	5 ab	11 abc	15 abcd
2 b	6 ac	12 abd	
3 c	7 ad	13 acd	
4 d	8 bc	14 bcd	
	9 bd		
	10 cd		

If a searcher were interested in a document containing any two of the descriptors above, say a and c, he would be led, as in a permutation index, to this document even though it contained two additional descriptors. Sharp (Sharp,66) observes that a user searching for attributes ac would be satisfied by the term acd or "any entry consisting of or beginning with the sought terms." The term ac is superfluous as are any entries contained in any larger entry; consequently, a further reduction of index entries

can be permitted. Terms 1,2,3,5,6,8, and 11 can be eliminated leaving:

1 d	2 ad	5 abd	8 abcd
	3 bd	6 acd	
	4 cd	7 bcd	

This is the absolute minimum number of entries required to still provide all coordinate entries. Since the indexing

---

Document Descriptors

13	ADP-BINDING-LIBRARY-SERIALS
21	ADP-CIRCULATION-LIBRARY-SCIENTIFIC
39	ADP-BIBLICS-LIBRARY
495	ACQUISITION-ADP-LIBRARY

Index Entries

ACQUISITION-ADP-LIBRARY-495
ACQUISITION-LIBRARY-495
ADP-BIBLICS-LIBRARY-39
ADP-BINDING-LIBRARY-SERIALS-13
ADP-BINDING-SERIALS-13
ADP-CIRCULATION-LIBRARY-SCIENTIFIC-21
ADP-CIRCULATION-SCIENTIFIC-21
ADP-LIBRARY-39
ADP-LIBRARY-495
ADP-LIBRARY-SCIENTIFIC-21
ADP-LIBRARY-SERIALS-13
ADP-SCIENTIFIC-21
ADP-SERIALS-13
BIBLICS-LIBRARY-39
BINDING-LIBRARY-SERIALS-13
BINDING-SERIALS-13
CIRCULATION-LIBRARY-SCIENTIFIC-21
CIRCULATION-SCIENTIFIC-21
LIBRARY-39
LIBRARY-495
LIBRARY-SERIALS-13
LIBRARY-SCIENTIFIC-495
SERIALS-13
SCIENTIFIC-495

Figure 3.1 A portion of a SLIC index

---

function generates a subset of all combinations of index terms, Sharp has dubbed this method Selected Listing In Combination (SLIC) as shown in Figure 3.1.

It is interesting to note that the only terms remaining are those combinations which end with the last descriptor of the assigned sequence. This simplifies the calculation of the total number of index entries to be entered in the index. If the final term (d in the example above) is dropped from each index entry, what remains is the sum of all combinations of n-1 items taken 0 through n-1 times, or

(i=0,n-1) SUM (c<n-1,i>) = 2**(n-1)

Algorithms for generating SLIC indexes have been given by Sharp {Sharp,66} and by Rush and Russo {Rush,71}.

SLIC techniques reduce the size of permuted indexes and retain coordinating ability yet still suffer from a multiplicity of entries when the number of assigned terms is large. The SLIC method produces 512 entries for a document assigned 10 terms: too many for some real applications.

3.1.4. PERMUTERM Index

Garfield {Garfield,55} has described an indexing function which compromises some coordinating ability for space. Uniterms assigned to a document form two distinct classes: main terms, which constitute the primary access points to the document; subordinate terms, modifying words which specify more clearly the sense in which a main term is used. For each main term, an index entry is constructed for

each of the remaining uniterms assigned to the document as a coordinate main-subordinate entry.

Assuming that the uniterms a and b are main terms of a document assigned concepts a,b,c, and d, then the index entries so generated are:

ab
ac
ad
ba
bc
bd

A PERMUTERM index collects in one place all subordinate entries, alphabetically ordered, pertaining to each main term found in a document collection. The indexing function approximates a subset of a completely permuted index (see section 3.1.2) whose entries are the permutations of all terms taken two at a time. Of n terms assigned to a document, assume m, $1 \leq m \leq n$, form the subset of main terms. The number of entries generated for this document is:

$$k * n * (n - 1) = k * n * (n - 1), \text{ for } 1/n \leq k \leq 1$$

When k maintains its average over its uniform interval of definition, then the number of entries generated per document is

$$(n^{**}2 - 1)/2$$

As employed by Garfield at the Institute for Scientific Information, the PERMUTERM index could be classified as a computer-generated index, discussed in more detail in the next section. Documents are assigned keywords extracted

from machine readable natural language titles. Single word concepts as well as frequently encountered word pairs matched from pre-compiled tables may be selected as main terms. Subordinate terms are automatically determined from a list of commonly applied modifiers. Figure 3.2 displays an example of a PERMUTERM index derived from the document descriptors of Figure 3.1.

ACQUISITIONS	CIRCULATION
ADP-495	ADP-21
LIBRARY-495	LIBRARY-21
ADP	SCIENTIFIC-21
ACQUISITION-495	LIBRARY
BIBLIOS-39	ADP-13
BINDING-13	ADP-21
CIRCULATION-21	ADP-39
LIBRARY-13	ADP-495
LIBRARY-21	ACQUISITION-495
LIBRARY-39	BIBLIOS-39
LIBRARY-495	BINIDNG-13
SCIENTIFIC-21	CIRCULATION-21
SERIALS-13	SCIENTIFIC-21
BIBLIOS	SERIALS-13
ADP-39	SERIALS
LIBRARY-39	ADP-13
BINDING	BINDING-13
ADP-13	LIBRARY-13
LIBRARY-13	SCIENTIFIC
SERIALS-13	ADP-21
	CIRCULATION-21
	LIBRARY-21

Figure 3.2 A portion of a PERMUTERM index

3.2. Computer-Generated Indexes

The preceding section has dealt with useful, automated means of displaying index terms once they have been associated with a document. This section examines the more

fundamental question of automatically selecting index terms from documents of natural language text.

Since derivative indexing techniques employ extractions from the document, the index descriptors must exist as some unit of the document itself. The most natural units of textual data are words or collections of words which form the objective index terms.

The underlying question which separates the techniques to be described is which words or phrases are to be chosen as representatives of the document and placed in the index. Of course one could easily argue that the ideal representative of a document, thus its ideal index entry, is the document itself. The indexing function in this case would do nothing but rearrange the units of the document and pass them to the index. The size of the index would be the sum of the sizes of the documents of the collection. The usefulness of such an index is doubtful since all units found in each document would be present in the index regardless of their importance to the subject matter discussed. Therefore, without some means of selectively choosing extractions from documents, computer-generated indexes would be of little value.

The selection of words or phrases naturally divides the index units of a document into two classes: those to be included in the index descriptor and those that are inappropriate as document representatives. Several ordering

relations are commonly applied to include or exclude units from these sets. A word could be chosen because of its form or position in a document - e.g. it may be included as an index entry if the word is capitalized and does not begin a sentence. The words themselves may be used as a clue - e.g. a word is indexable if it isn't non-indexable (this stoplist technique of admitting index entries will be discussed in section 3.2.1). Or, the statistical nature of the document can describe its own descriptors - e.g. the ten most frequently found non-common words of the document can be chosen.

3.2.1. Key-Word-in-Context (KWIC) Index and Key-Word-out-of-Context (KWOC) Index

In striving for a speedy, totally automated method of index construction, H. P. Luhn reasoned that the organization of index entries must rely on terms extracted from an author's text rather than assigned in accordance with human judgement {Luhn,59}. The simplest form of such an index might be an alphabetic listing of keywords found in a document; however, to insure the proper meaning of such keywords, the user would have to refer to the text from which the word was extracted. To alleviate this tedious procedure, Luhn proposed listing selected "keywords together with surrounding words acting as modifiers to specify the sense in which the keyword was applied". The added degree of keyword specification by such key-word-in-context, KWIC,

indexes is easily accomplished by automatic means.

The keywords of a document need only be defined as those words which characterize a subject more than others. Since word significance is often difficult to precisely define, it becomes more practical to reject all obviously non-significant words, retaining any others as significant with the risk of admitting words of questionable status. A list of these non-significant words, called a stoplist, would include prepositions, conjunctions, articles, auxilary verbs, certain adjectives, and words of little informative value such as "report", "theory", and the like.

Computer-generated KWIC indexes have become an important tool in the maintenance of truly current awareness because of the speed and simplicity of the indexing method. The text of an author's title, a sentence from an abstract, or full text is submitted in machine readable form. Each word of the text is processed against the stoplist eliminating words found therein from further processing. The remaining presumably significant words are rotated, one at a time in succession, to an indexing position or keyword window where a snapshot of the keyword and its surrounding context is recorded. This process is repeated until all the text of the collection has been submitted. The recorded images are then alphabetically arranged by the keyword appearing in the indexing position and listed with as much surrounding context as will fit within a column on the

printed output page.

Since its introduction by Luhn {Luhn,59} and Citron {Citron,59}, the KWIC index has taken on many display formats, each claiming to have certain advantages. The most common, shown in Figure 3.3, displays on a single line a centrally located keyword with the surrounding context "wrapped around" to present the user with as much of the modifying phrase regardless of the location of the keyword in the sentence. This format leads the user directly to the keyword window allowing him the freedom to browse in the modifying context upon locating a keyword of interest. When the context following the keyword is used to further order index entries, multi-word phrases beginning with the same keyword are clumped together providing limited search capabilities for more specific concepts. However, all valid coordinations of words producing this multi-word concept are

---

```
Y CYLINDERS AT LIQUID + FLUX JUMPS IN NICBIUM-ZIRCONIUM ALLO
LE FOR A LUBRICANT IN A FLUX OF SULFURIZING GASES.= +SUITAB
                        FOAM FRACTIONATION OF POLYMERS.=
ING IN NUCLEAR HARTREE- FOCK ORBITALS AND ELASTIC AND QUASIF
. RELATION TO HARTREE- FOCK THEORY.= +ATOMIC POLARIZABILIT
DGE ELECTRON P+USE OF A FOCUSSING SPECTROMETER WITH A CAMBRI
ND DIALYZED EXTRACTS OF FODDER AND BAKER'S YEASTS.= +A
APHYLOCOCCAL NUCLEASE ( FOGGI STRAIN). ORDER OF CYANOGEN
 CONDUCTIVITY OF COPPER FOIL AT LOW TEMPERATURES.= +ON THE
BI CRYSTALLINE ALUMINUM FOIL.=+ MIGRATION PHENOMENA IN THIN
ECIUM. PURIFICA+DIHYDRO FOLATE REDUCTASE OF STREPTOCOCCUS FA
OPERTIES OF TWO DIHYDRO FOLATE REDUCTASES FROM THE AMETHO
TION PRODUCT OF DIHYDRO FOLATE.=+IDENTIFICATION AS A DEGRADA
THE RHIZOSPHERE EFFECT. FOLIAR APPLICATION OF CERTAIN CHEMIC
```

Figure 3.3 A portion of a KWIC index

---

not necessarily located at that position in the index since the secondary descriptor may be located at some point other than the word immediately following the keyword. Thus, to locate those scattered, more specific concepts, the entire text of all titles containing this keyword must be scanned to spot all occurrences of secondary descriptors. When a significant word appears frequently in the indexed text, this format may discourage users from the sequential scanning of long blocks of identical keywords.

Many users of these indexes were unsatisfied with the KWIC format, having been accustomed to the more traditional forms of subject indexes. To satisfy these users, a variation of the KWIC indexing method generates subject headings by extracting the keyword from the context forming a keyword-out-of-context (KWOC) index as shown in Figure 3.4. In this figure each KWOC index entry retains the entire text of the title or phrase from which the keyword was extracted. Other variations may include only a portion of the title or phrase from which the keyword was extracted. Coordinate searches are difficult to perform in these indexes since no subordering scheme is employed to collate secondary concepts. Thus, the user is forced to linearly scan each title phrase posted beneath the extracted term for secondary concepts of interest.

The single, flexible determinant of the quality and the size of KWIC index lies in the words found on the stoplist.

DNA

ISOLATION OF * AND RIBOSOMAL RNA FROM RAT LIVER.=	246
* BASIC COMPOSITION OF HUMAN T-STRAIN MYCOPLASMS.=	44
PHOTOPRODUCTS IN * IRRADIATED IN VIVO.=	643
TURNOVER OF NUCLEAR * LIKE RNA IN HELLA CELLS.=	112
EFFECTS OF METALS ON THE MECHANISM OF ACTIVATED * NUCLEASES.=	402
USE OF A NEW METHOD TO OBSERVE THE KINETIC REACTIONS OF * NUCLEASES.=	401
DENATURATION MAP OF POLYOMA VIRUS * .=	242
ELECTRIC CONDUCTIVITY OF SODIUM SALTS OF * .=	648
EFFECT OF SOME MUTAGENIC VIROGENS AND CARCINOGENS ON * .=	131

DOCOSAHEXAENOIC

PREDICTING THE POSITIONAL DISTRIBUTION OF * AND DOCOSAPENTAENOIC ACIDS IN ANIMAL TRI GLYCERIDES	417

Figure 3.4 A portion of a KWOC index

Short lists, rejecting only the most obvious insignificant words, admit many index terms of doubtful value and needlessly increase the size of the final index. The general subject matter of a corpus of documents dictates, to a great degree, word usage. The vocabulary of chemistry differs greatly from that of mathematics. Stoplists constructed for preparing indexes of document collections from these fields could be expected to be similar only at the most common word level comprising conjunctions,

articles, and a few adjectives. Words which could be highly relevant to one subject area may be so common or uninformative to another field as to appear on the index construction stoplist of the latter. For example, the word "field" carries a strict definition within mathematical disciplines, while in agriculture, the word has little significance. Placing a word on the stoplist which could generate many index entries is a common practice which reduces "block fatigue" and size on the one hand, but totally denies user access through this word on the other! The economic balance of the number of lines to be printed against the loss of retrieval effectiveness if words are omitted from the search is the critical question that must be decided in the establishment of stoplists.

To estimate the size of a KWIC index, the relative number of non-significant words must be estimated as well as the average number of words per document. If p is the fraction of significant words of a n-word document, then the breadth of indexing is p * n. Most KWIC indexes and some KWOC indexes require one line per entry; thus, the number of lines in an index of m documents is m * p * n. The size of a KWOC index is approximately the same as that produced by KWIC indexing methods when the title is printed on a single line. When the full title phrase is presented in the index, the size estimates become more data base dependent.

The KWIC and, to a greater extent, the KWOC indexes suffer from limitations of not allowing one to perform, easily, an arbitrary coordinate search when large numbers of entries are posted with the same keyword. In general, each KWIC or KWOC entry must be linearly scanned for any secondary concepts. If, in a KWIC index, the secondary term immediately follows the primary keyword, then these entries are collected in that place in the index (see Figure 3.3, FOLATE REDUCTASE). However, all other coordinations of terms are randomly scattered to the left or right of the primary posting.

3.2.2. PANDEX Index

A relatively recent form of automatic indexing, known as PANDEX and published by CCM Information Corporation {CCM,72}, incorporates term coordination in an interesting variation of a KWOC index. Keywords are extracted from titles as in a KWOC index. The entire text of the title is posted as a subordinate entry ordered alphabetically by a secondary keyword found in the context at close proximity to the extracted term. Both primary and secondary keywords are printed in boldface to attract the user's eye as demonstrated in Figure 3.5.

Depending upon the nature of the surrounding context, the boldface term constitutes a more specific concept by adding a significant word from either the right or left of the main term. Assume that w<0> is the primary keyword

selected. The title may then be stylized as

...w<-3> w<-2> w<-1> w<0> w<1> w<2> w<3> ...

where w<i> represents a word of the title and i its position relative to the primary keyword. The subordinate term is immediately chosen if w<1> is a significant keyword (i.e. w<1> is not in the stoplist). Otherwise, the subordinate concept is sought by examining w<-1>. If this word too is on the stoplist, w<-2> is examined. Reasoning that w<-1> may be a function word such as "of", "in", "on", etc., w<-2> is functionally related to w<0> producing a relevant concept coordination. If w<-2> is non-indexable, the secondary keyword is sought alternately from the right and left of the keyword position. The chosen secondary keyword is then the first indexable word of the sequence

w<1> w<-1> w<-2> w<2> w<-3> w<3> ...

The phrase being indexed has first and last words. Consequently, some members of the above sequence may be nonexistent. The PANDEX construction algorithm further restricts the range of the secondary keyword search by bounding the words examined by certain punctuation found in the title. A colon, semicolon, or period indicate the introduction or termination of concepts within an index phrase. By limiting secondary keywords to these bounded subphrases, more useful coordinate terms are chosen.

Although the keywords are printed in boldface, the user must still locate them within the title which may cause as

THYROID
Effect of propyl thio uracil in the survival of rat THYROID CELLS in vivo and in vitro.= 577

Thyro Globulin Immunity. Effect of THYROID IMMUNE and other protein-thyroxine complexes on tissue concentration of labeled thyroxine and tadpole metamorphosis.= 71

THYROXINE
Thyro Globulin Immunity. Effect of Thyroid Immune and other protein THYROXINE COMPLEXES on tissue concentration of labeled thyroxine and tadpole metamorphosis.= 71

THYROXINE DEGRADATION. Anti-oxidant function and non-enzymic degradation during microsonial lipid per oxidation.= 91

Figure 3.5 A portion of a PANDEX index

much duress as scanning large blocks of KWIC entries. He may well have to scan the entire block containing a keyword of interest anyway since only one extra keyword is highlighted. The user may find clues from other words of the phrase.

3.2.3. Articulated Subject Index

The organization of both the KWIC and KWOC indexes lead a user to perform much unnecessary scanning of irrelevant context surrounding keywords. PANDEX, to some extent, overcomes this problem though still not adhering to the organizational structure of subject indexes or back-of-the-book indexes.

The automatic generation of subject indexes from title-like phrases has been studied by Armitage and Lynch from

examinations of the subject index to Chemical Abstracts {Armitage,67}. The articulated subject index consists of a set of subject headings, in alphabetical order, under which are indented a series of modifying phrases or modifiers (see Figure 3.6). The modifiers are listed in alphabetical order by their significant words. Common words such as prepositions, conjunctions, and articles are ignored when ordering the modifiers.

```
Cesium
  absorption by plants, fertilizer effect on, 60:13833f
    by plants, soil colloids and, 60:11321h
    by roots, Ca and, 60:12620b
  adenosine triphosphatase response to, 60:4400b
  adsorption of, by Hg electrodes, in presence of methyl-
    formamine, 60:8668c
    from radioactive waste water by clay, 60:3865e
    from Na soln. by clinoptilolite, heat-treatment effect
      on, 60:15482h
  argoid gel properties in presence of, 60:6246e
  atomic scattering factor of, 60:7528d
  from barium-133 decay, angular correllation, 60:1283h
  base exchange of, in alcs. and aq. alcs., 60:2359b
    with ammonia on fajasite-type zeolites, 60:7490h
    on Bio-Rex 70 and Dowex-50W, hydration in relation to,
      60:42e
    with Ca and Li solvents in relation to, 60:7493c
    with K and Na in zeolites, 60:13024g
    with Na in two-temp. process, 60:9951d
```

Figure 3.6 A portion of an Articulated Subject Index.

A subject heading, together with its modifiers can be arranged to form a meaningful phrase. The method of synthesizing this descriptive phrase from an index entry provides a basis for automatic construction of the index

entries. Some of the very words found on stoplists for KWIC index construction - prepositions and conjunctions - separate the full phrase into substantive phrases which can act as subject headings in an index. A full phrase can be represented as a string of n substantive phrases separated by n-1 function words (articulation points):

- c - o - o - o - o -

where

- indicates a substantive phrase

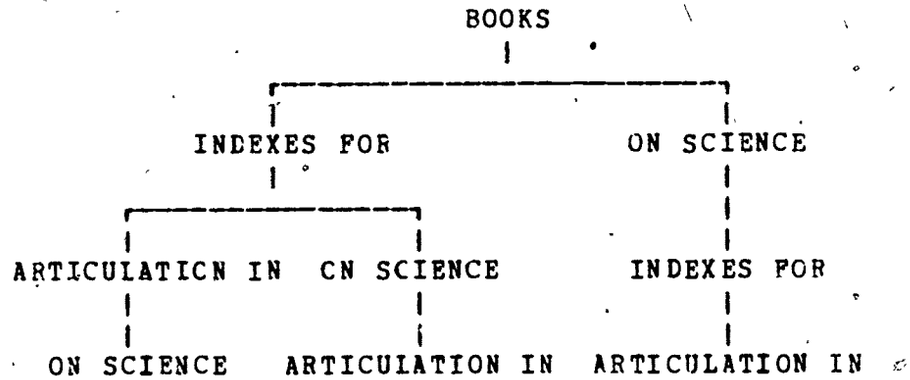
o indicates an articulation point

The modifiers may be further broken into components and separated by commas. When two or more modifiers share an initial component, the component is printed once and the remaining modifiers are indented beneath. In this manner, a high degree of organization is introduced into the index display permitting useful coordination of components with the subject heading.

This model of an articulatable phrase serves as the simplest example for the logical generation of subject headings. The general rule for constructing index entries states that if one of the substantive phrases is chosen as a subject heading, then all possible modifiers are formed by choosing an adjacent function word and subphrase adjacent to it and continuing this selection as long as the first subphrase has not been chosen. At each stage, sets of contiguous function-word-phrases may be chosen. For

example, if "books" is chosen as a subject heading, the selection of modifiers is given in the example below:

ARTICULATION IN INDEXES FOR BOOKS ON SCIENCE



In standard form,

BOOKS
INDEXES FOR, ARTICULATION IN, ON SCIENCE
INDEXES FOR, ON SCIENCE, ARTICULATION IN
ON SCIENCE, INDEXES FOR, ARTICULATION IN

The multiple set "ARTICULATION IN INDEXES FOR" could have been chosen yielding the added terms

ARTICULATION IN INDEXES FOR, ON SCIENCE
ON SCIENCE, ARTICULATION IN INDEXES FOR

All possible index entries for this phrase are illustrated in Figure 3.7.

To reconstruct the full descriptive phrase from an index entry simply concatenate the components, in the order specified by the modifier, to the left of the subject heading if the component ends with a function word, or to

ARTICULATION
IN INDEXES FOR BOOKS ON SCIENCE

BOOKS
ARTICULATION IN INDEXES FOR, ON SCIENCE
INDEXES FOR, ARTICULATION IN, ON SCIENCE
———, ON SCIENCE, ARTICULATION IN
ON SCIENCE, ARTICULATION IN INDEXES FOR
———, INDEXES FOR, ARTICULATION IN

INDEXES
ARTICULATION IN, FOR BOOKS ON SCIENCE
FOR BOOKS ON SCIENCE, ARTICULATION IN
FOR BOOKS, ARTICULATION IN, ON SCIENCE

SCIENCE
ARTICULATION IN INDEXES FOR BOOKS ON
BOOKS ON, ARTICULATION IN INDEXES FOR
———, INDEXES FOR, ARTICULATION IN
INDEXES FOR, ARTICULATION IN BOOKS ON

Figure 3.7 All articulated index phrases generated from the title "Articulation in Indexes for Books on Science"

the right if the component begins with a function word.

Articulated subject indexes are perhaps the most useful that could be constructed from single title-like phrases by strictly derivative techniques. The depth of an articulated subject index equals that of any other indexing method previously discussed. Its power lies in the organization and depth of the entries. Coordination of subphrases can be performed to the limit of discriminating among any similar phrases, regardless of the position of the subject heading or components within the full phrase. The size of the index, though somewhat large when compared to KWIC (see appendix A), could possibly be tolerated when its usefulness

is considered. Armitage and Lynch {Armitage,67} have presented several rules for trimming the number of entries generated per phrase, claiming to retain all useful coordinations.

The major drawback to the articulated subject index approach is the English language itself. Not all title phrases follow the simple model of an articulated phrase. In their study of Chemical Abstracts, Armitage and Lynch found that only 66% of the phrases examined conformed to this "normal form" {Armitage,67}. The most common causes for irregularities were:

a) use of adjectival modifiers instead of articulated phrases;

b) use of infinitives and other verb constructions.

To perform 100% of the time, as KWIC indexing methods do, automated articulated subject index construction must either resort to automated syntactic analysis of natural language text or a manual editing of titles presented for input.

The first alternative is desirous since many commercial institutions are providing document titles in computer readable form. An in depth syntactic analysis of titles would permit the entire indexing process to continue automatically. On the other hand, to be competitive costwise with KWIC techniques, the computing time should be minimized - a highly improbable task when analyzing natural

text.

Manual editing of titles is equally undesirable. Trained indexers would undoubtedly be necessary to perform such tasks, interjecting error, inconsistency, and cost to the indexing procedure.

Young and Rush {Young,72} are examining the problems of automatically "normalizing" phrases through linguistic analysis so that articulated subject index algorithms can be directly applied.

3.3. Approach Explored in this Thesis

The approach to improved index construction explored in this thesis combines many of the aspects of computer-generated and computer-compiled techniques. The discussion and illustrations of section 3.2.1 have demonstrated the capabilities of the KWIC indexing technique to provide immediate access to all significant words of a title; however, secondary concepts must be found by searching for contextual relationships in the text about the keyword. The PERMUTERM index, discussed in section 3.1.4, provides immediate access to secondary concepts; however, since no syntax information is supplied concerning the relationship between the subordinate and main keywords, false retrievals may occur when the concepts described by these single keywords are not related in the same manner expected by a user.

The chapters to follow discuss a refinement of the KWIC indexing technique which combines the immediate secondary access capabilities of the PERMUTERM indexing technique and the contextual relationships and automated construction ease of the KWIC indexing technique to produce indexes which approach the usefulness of articulated subject indexes.

CHAPTER IV. THE PROTOTYPE DOUBLE-KWIC (DKWIC) COORDINATE INDEX

The need for high-quality printed indexes to facilitate manual retrieval of information has not diminished, despite the strides that have been made in the development of automatic information retrieval systems. Nevertheless, attempts to produce high-quality indexes by automated techniques have only recently begun to merit serious attention (see Chapter 3). Perhaps the most significant breakthrough in this area occurred when Luhn and others successfully applied the key-word-in-context (KWIC) indexing concept as an automated indexing technique (see section 3.2.1). The widespread use of KWIC indexes since that time and the variety of formats in which they have appeared have been reviewed by Fischer {Fischer,66} and others {Adams,68,Stevens,65}.

The rapid rise in popularity of KWIC indexes apparently has been due to the high speed and low cost of producing them. However, as noted by Fischer, there has been some dissatisfaction with the quality of KWIC indexes. Most of the attempts to improve quality have dealt with variations in format to improve readability or with enrichment of titles to provide additional index entries which otherwise would not have been derived from words in the titles.

The enrichment of titles improves the quality of KWIC

indexes by increasing the breadth of indexing. An equally attractive possibility, which appears to have been little explored, involves extension of the KWIC indexing principle to provide for an increased depth of indexing. If a greater depth of indexing were possible, it would help to overcome one of the major drawbacks of KWIC indexing, namely, searching for a specific concept when a large number of index entries are posted under a given keyword.

One of the difficulties encountered in such a situation is illustrated by the set of KWIC index entries shown in Figure 4.1 which are taken from a KWIC index of titles from

TION OF STRUCTURAL INFORMATION.=+STORAGE AND VERIFICA	43
E COMMUNICATION OF INFORMATION.=+ARCH RELATING TO THE	B257
YBOARDING CHEMICAL INFORMATION.= KE	232
OR A LA+ SELECTIVE INFORMATION ANNOUNCEMENT SYSTEMS F	142
+REVIEW: TECHNICAL INFORMATION CENTER ADMINISTRATION+	B257
+TEM AND AUTOMATIC INFORMATION DISTRIBUTION USING CO+	124
+TION OF TECHNICAL INFORMATION GROUPS - INTRODUCTORY+	110
ING + BOOK_REVIEW: INFORMATION MANAGEMENT IN ENGINEER	B2-2
ING AN OPERATIONAL INFORMATION PROGRAM.=+ORS IN BUILD	107
THE B.F. GOODRICH INFORMATION RETRIEVAL SYSTEM AND +	124
BASED + BIOMEDICAL INFORMATION RETRIEVAL: A COMPUTER-	98
+ ANNUAL REVIEW OF INFORMATION SCIENCE AND TECHNOLOG+	B3-2
INING PROGRAMS FOR INFORMATION SCIENTISTS.=+DEMIC TRA	118
ATION IN TECHNICAL INFORMATION SERVICES.=+INUING EDUC	115
ATION OF TECHNICAL INFORMATION SERVICES.=+ AND INTEGR	111
+EMICALLY ORIENTED INFORMATION STORAGE AND RETRIEVAL+	43
TORIAL: A NATIONAL INFORMATION SYSTEM.= EDI	E 61
RGE-SCALE CHEMICAL INFORMATION SYSTEMS.=+TION IN A LA	192
TERMINING COSTS OF INFORMATION SYSTEMS.= DE	101
+FIC AND TECHNICAL INFORMATION SYSTEMS IN CURRENT US+	B3-2

Figure 4.1 ? portion of a conventional KWIC index illustrating the randomizatior of secondary concepts found for a high-density keyword. Note the randomization of concépts "TECHNICAL INFORMATION", "INFORMATION STORAGE", and "INFORMATION RETRIEVAL".

Volume 7 of the Journal of Chemical Documentation. Because these index entries are subordered on the basis of words immediately following the word in the index column, the resulting order differs markedly from the usual order one would find in a back-of-the-book index or an articulated subject index. For example, several of the entries indexed under "INFORMATION" indicate that the titles deal with "TECHNICAL INFORMATION," but the entries are scattered because of the ordering principle just described. A similar situation applies to entries describing "INFORMATION RETRIEVAL" and "INFORMATION STORAGE" brought about by slight differences in title phraseology.

In another format for the KWIC index (Figure 4.2), a variant of the KWOC format discussed in section 3.2.1, the situation is even worse. In this format, the index word is extracted from the title and replaced by an asterisk to indicate its location in the title. All of the titles, or portions thereof, from which a given index term is extracted are then grouped together under that index term and are subordered on the basis of the accession numbers for the titles from which they are derived. This method of ordering is worse than the first, because of complete randomization of the words to the right as well as to the left of the index words. Also, this second format makes it more difficult to determine the immediate context about the keyword when scanning the individual entries, since the

```
INFORMATION
  SEM. 1. STORAGE AND VERIFICATION OF STRUCTURAL * .=      43
  A CHEMICALLY ORIENTED * STORAGE AND RETRIEVAL SYSTE      43
  BIOMEDICAL * RETRIEVAL: A COMPUTER-BASED SYSTEM FOR      98
  DETERMINING COSTS OF * SYSTEMS.=                        101
  FACTORS IN BUILDING AN OPERATIONAL * PROGRAM.=          107
  SYMPOSIUM ON ADMINISTRATION OF TECHNICAL * SERVICES     110
  COORDINATION AND INTEGRATION OF TECHNICAL * SERVICE     111
  CONTINUING EDUCATION IN TECHNICAL * SERVICES.=          115
  SALARIES AND ACADEMIC TRAINING PROGRAMS FOR * SCIEN     118
  THE B.F. GOODRICH * RETRIEVAL SYSTEM AND AUTOMATIC      124
  AUTOMATIC * DISTRIBUTION USING COMPUTER-COMPILED TH     124
  SELECTIVE * ANNOUNCEMENT SYSTEMS FOR A LARGE COMMUN     142
  NIQUE NOTATION IN A LARGE-SCALE CHEMICAL * SYSTEM.=     192
  KEYBOARDING CHEMICAL * .=                               232
  BOOK_REVIEW: * MANAGEMENT IN ENGINEERING EDUCATION.    B2-2
  BOOK_REVIEW: ANNUAL REVIEW OF * SCIENCE AND TECHNOL    B3-2
  CIENTIFIC AND TECHNICAL * SYSTEMS IN CURRENT USE.=     B3-2
  HY OF RESEARCH RELATING TO THE COMMUNICATION OF * .=   B3-2
  BOOK_REVIEW: TECHNICAL * CENTER ADMINISTRATION, VOL    B257
  EDITORIAL: A NATIONAL * SYSTEM.=                       E 61
```

Figure 4.2 A variant form of a KWIC (also called KWOC) index illustrating complete randomization of secondary concepts for the same titles illustrated in Figure 4.1

keyword - in this case, its identifying asterisk - no longer appears in a fixed position.

In another format for a KWOC index of these same titles (Figure 4.3), the keyword is extracted and the full text of the altered title is posted beneath this term. The subordering of altered titles is arbitrary, or as shown in Figure 4.3, the words following the extracted term are used. Although all concepts of the original title are retained, the randomization of words to the left of the index term as well as non-contiguously to the right forces the user of a KWOC index to scan all the text of each entry to identify

all articles describing a secondary concept.

INFORMATION

A CHEMICALLY ORIENTED INFORMATION STORAGE AND RETRIEVAL SYSTEM. 1. STORAGE AND VERIFICATION OF STRUCTURAL * .=	43
BOOK_REVIEW: BIBLIOGRAPHY OF RESEARCH RELATING TO THE COMMUNICATION OF * .=	B257
KEYBOARDING CHEMICAL * .=	232
SELECTIVE * ANNOUNCEMENT FOR A LARGE COMMUNITY OF USERS.=	142
BOOK_REVIEW: TECHNICAL * CENTER ADMINISTRATION, VOL 3.=	B257
SYMPOSIUM ON ADMINISTRATION OF TECHNICAL * GROUPS - INTRODUCTORY REMARKS.=	110
BOOK_REVIEW: * MANAGEMENT IN ENGINEERING EDUCATION .=	B2-2
FACTORS IN BUILDING AN OPERATIONAL * PROGRAM.=	107
B.F. GOODRICH * RETRIEVAL SYSTEM AND AUTOMATIC INFORMATION DISTRIBUTION USING COMPUTER-COMPILED THESAURUS AND DUAL DICTIONARY.=	124
BIOMEDICAL * RETRIEVAL: A COMPUTER-BASED SYSTEM FOR INDIVIDUAL USE.=	98
BOOK_REVIEW: ANNUAL REVIEW OF * SCIENCE AND TECHNOLOGY.=	B3-2
SALARIES AND ACADEMIC TRAINING PROGRAMS FOR * SCIENTISTS.=	118
CONTINUING EDUCATION IN TECHNICAL * SERVICES.=	115
COORDINATION AND INTEGRATION OF TECHNICAL * SERVICES.=	111
A CHEMICALLY ORIENTED * STORAGE AND RETRIEVAL SYSTEM. 1. STORAGE AND VERIFICATION OF STRUCTURAL INFORMATION.=	43
EDITORIAL: A NATIONAL * SYSTEM.=	E 61
USE OF NONUNIQUE NOTATION IN A LARGE-SCALE CHEMICAL * SYSTEM.=	192
DETERMINING COSTS OF * SYSTEMS.=	101
BOOK_REVIEW: NONCONVENTIONAL SCIENTIFIC AND TECHNICAL * SYSTEMS IN CURRENT USE.=	B3-2

Figure 4.3 Another KWOC format illustrating complete randomization of secondary concepts for the high-density concepts of Figure 4.1

The PANDEX format (see section 3.2.2) for these same titles (Figure 4.4) leaves something to be desired also. The PANDEX index construction generally performs a coordination of a single secondary concept with the main index term from a given title. The title, however, may contain other secondary concepts not highlighted in the index phrase. In many instances, the secondary concept chosen does not represent the most appropriate subordinate term. The selection of subordinate concepts can induce further scattering of terms. Four occurrences of the phrase "TECHNICAL INFORMATION" appear in the titles indexed in Figure 4.4, yet only two entries specify "TECHNICAL" as the highlighted concept. To locate all occurrences of a more specific concept, a user will be forced to linearly scan the text of all titles posted beneath the main heading much as in a KWIC or KWOC index.

To overcome some of the difficulties of these indexing approaches, studies have been initiated by Armitage and Lynch [Armitage,67], Dolby [Dolby,68], and others to analyze the characteristics of traditional subject indexes. Their approaches tend to require linguistic analysis of titles and title-like phrases to effect the transformations required to produce such higher-quality indexes by automated techniques (see section 3.2.3). This chapter presents a more simplified approach to automatic preparation of higher-quality indexes, based on an extension of the KWIC indexing

INFORMATION
- Keyboarding CHEMICAL INFORMATION.= 232
- Book_review: Bibliography of Research Relating to the COMMUNICATION of INFORMATION.= B257
- B.F. Goodrich Information Retrieval System and automatic INFORMATION DISTRIBUTION using Computer Compiled Thesaurus and Dual Dictionary.= 124
- Book_review: INFORMATION MANAGEMENT in Engineering Education.= B2-2
- Factors in Building an Operational INFORMATION PROGRAM.= 107
- Biomedical INFORMATION RETRIEVAL: A Computer-based System for Individual Use.= 98
- B.F. Goodrich INFORMATION RETRIEVAL System and Automatic Information Distribution using Computer-Compiled Thesaurus and Dual Dictionary.= 124
- Book_review: Annual Review of INFORMATION SCIENCE and Technology.= B3-2
- Salaries and Academic Training Programs for INFORMATION SCIENTISTS.= 118
- SELECTIVE INFORMATION Announcement for a large Community of Users.= 142
- Coordination and Integration of Technical INFORMATION SERVICES.= 111
- Continuing Education if Technical INFORMATION SERVICES.= 115
- A Chemically Oriented INFORMATION STORAGE and Retrieval System. 1. Storage and Verification of Structural Information.= 43
- Determining Costs of INFORMATION SYSTEMS.= 101
- Use of Nonunique Notation in a large-scale Chemical INFORMATION SYSTEM.= 192
- Editorial: A National INFORMATION SYSTEM.= E 61
- Book_review: Nonconventional Scientific and Technical INFORMATION SYSTEMS in Current Use.= B3-2
- Symposium on Administration of TECHNICAL INFORMATION Groups - Introductory Remarks.= 110
- Book_review: TECHNICAL INFORMATION Center Administration, Vol 3.= B257

Figure 4.4 A PANDEX index for the same titles of Figure 4.1 illustrating partial ordering of a single secondary concept for each title where the secondary concept chosen is not always the most appropriate one

concept. For reasons which will soon become apparent, we have chosen the name "Double-KWIC Coordinate Index" for the printed index produced by this new approach.

4.1. Construction of The Double-KWIC Coordinate Index

As illustrated in Figure 4.5, the double-KWIC coordinate index is constructed as follows:

1) The first significant word in a title is extracted as a main index term and replaced by an asterisk (*) to indicate its position in the title.

2) The remaining words in the title are then rotated, so as to permit each significant word to appear as the first word of a wrap-around subordinate entry under the main index term.

Steps 1 and 2 are repeated until all of the titles of a given bibliographic listing are processed. The index entries so created are then sorted alphabetically, both with regard to main terms (primary sort) and subordinate terms (secondary sort). Word significance for selection of main index terms and subordinate index terms is established on the basis of stoplists, discussed later. Also, main index terms are not restricted to single words, but may consist of multi-word terms derived from contiguous sets of words in the titles.

To illustrate some of the advantages of the double-KWIC coordinate indexing technique and to provide some comparison with indexing schemes described and illustrated in the

```
                                                        ┌──TITLE
                                                        │
    THE NOMENCLATURE OF HIGHLY FLUORIDATED MOLECULES.=  25

MAIN                                  MAIN TERM EXTRACTED
TERM                                          │
   └──────>NOMENCLATURE                       │
             FLUORIDATED MOLECULES.= THE * OF HIGHLY  25
             HIGHLY FLUORIDATED MOLECULES.= THE * OF  25
          ┌──> MOLECULES.= THE * OF HIGHLY FLUORIDATED  25
SUBORDINATE
  TERM       MOLECULES
             NOMENCLATURE OF HIGHLY FLUORIDATED * .=  25
             NOMENCLATURE OF HIGHLY FLUORIDATED *.=   25
             FLUORIDATED * .= NOMENCLATURE OF HIGHLY  25

           FLUORIDATED MOLECULES
             NOMENCLATURE OF HIGHLY * .=  25
             HIGHLY * .= NOMENCLATURE OF  25
```

Figure 4.5 Construction of the prototype double-KWIC (DKWIC) coordinate index entries

introduction to this chapter, a prototype DKWIC index was prepared (Petrarca,69a) from the same titles used for creating those sample illustrations (i.e., those titles appearing in Volume 7 of the Journal of Chemical Documentation). The prototype index was derived from 71 titles and contained approximately 1500 primary and secondary access points. A KWIC index prepared from these same titles contained only 350 primary access entries.

Figure 4.6 illustrates an annotated portion of the display format used for the prototype index produced by the double-KWIC coordinate indexing scheme discussed above. The complete prototype index has been published elsewhere (NAPS,69).

```
          1              5                    4        7
BOOK_REVIEW<-J           |                    |        |
  ADMINISTRATION, VOL 3.= +CHNICAL INFORMATION CENTER    B257
  ANALYSIS.= +NG NUMERICAL DATA PROJECTS A SURVEY AND    B2-2
  ANALYSIS, VOL 4.= +YCLOPEDIA OF INDUSTRIAL CHEMICAL    B258
  ANNUAL REVIEW OF INFORMATION SCIENCE AND TECHNOLOG+     B3-2
  APPLICATIONS.=..... *: COMPUTER                        B258
  BASIC PRINCIPLES OF CHEMISTRY.=..... *:                 B3-2
  BIBLIOGRAPHIC REVIEW.=.. *: SALICYLATES. A CRITICAL     B 59
  BIBLICGRAPHIC, AND CATALOG ENTRIES.= +ING OF INDEX,    B2-2
  BIBLIOGRAPHY OF RESEARCH RELATING TO THE COMMUNICA+    B257
  BIOCHEMICAL PREPARATIONS.=..... *:                      B3-2
  BOOK.=..... *: CHEMICAL DATA                           B2-2
  BOOK OF CHEMISTRY.=..... *: REFERENCE                  B2-2
  CAS TODAY.=..... *:                                    B182
  CATALOG ENTRIES.= +ING OF INDEX, BIBLICGRAPHIC, AND    B2-2
  CENTER ADMINISTRATICN, VOL 3.= +CHNICAL INFORMATION    B257
  CHEMICAL ANALYSIS, VCL 4.= +YCLOPEDIA OF INDUSTRIAL    B258
  CHEMICAL DATA BOOK.=..... *:                           B2-2
     |              |                    |
     3              6                    2
```

1 - Main index term
2 - location of main index term in title being permuted (rotated) for creation of subordinate entries.
3 - subordinate index term
4 - word in wrap-around title which immediately precedes subordinate index term
5 - truncation symbol used when words in wrap-around title do not fit in alloted field
6 - symbol indicating the end of title
7 - accession code for title represented by subordinate phrase. Alphabetic characters preceding the page number represent the following: B - book review; E - editorial. Also, the two page-numbering systems used by the Journal are represented by the following formats: (1) Unhyphenated - arabic numbered pages used for sequential numbering of the pages for Volume 7; (2) Hyphenated - Roman numeral pages for the individual issues of Volume 7. The number preceding the hyphen is the issue number.

Figure 4.6 Annotated description of the display format for the prototype double-KWIC coordinate index derived from titles in Journal of Chemical Documentation, Volume 7

4.2 Utility of the Double-KWIC (DKWIC) Coordinate Index

To illustrate some of the advantages of the double-KWIC coordinate indexing technique, Figures 4.7 through 4.9 display portions of the prototype DKWIC index for comparisons with portions of the indexes shown in Figures 4.1 through 4.4 which were derived from the same titles. Figure 4.7 illustrates the portion of the DKWIC index for the main term "INFORMATION". The DKWIC index eliminates the random ordering of subordinate concepts found in the KWIC index and its variants (Figure 4.1 - 4.3). The alphabetic ordering of subordinate concepts of the DKWIC construction enables one to quickly scan the subordinate index terms to find the particular subordinate concept. Since all significant words remaining in the titles are chosen as subordinate terms, all secondary terms chosen for the PANDEX index are included in the DKWIC index. Note that in the DKWIC index all titles pertaining to "TECHNICAL INFORMATION" are located in one place (see Figure 4.7).

Both the KWIC and DKWIC indexes would permit one to locate equally as well those precoordinate index terms under the heading for the modifier immediately preceding the word "INFORMATION". The PANDEX index aids in this coordination by highlighting some of these important words as noted in Figure 4.4. However, as illustrated in Figure 4.8, the DKWIC index permits immediate access to these precoordinate entries through the creation of multi-word main terms.

```
SELECTIVE * ANNOUNCEMENT FOR A LARGE COMMUNITY OF *    142
STORAGE AND RETRIEVAL SYSTEM. 1. STORAGE AND VERIF+     43
STORAGE AND VERIFICATION OF STRUCTURAL * .= +EM. 1.     43
SYMPOSIUM ON ADMINISTRATION OF TECHNICAL * GROUPS +     43
SYSTEM.= +IQUE NOTATION IN A LARGE-SCALE CHEMICAL *    192
SYSTEM.=.....EDITORIAL: A NATIONAL *                   E 61
SYSTEM. 1. STORAGE AND VERIFICATION OF STRUCTURAL +     43
SYSTEM AND AUTOMATIC * DISTRIBUTION USING COMPUTER+     124
SYSTEMS.=.....DETERMINING COSTS OF *                    101
SYSTEMS IN CURRENT USE.= +CIENTIFIC AND TECHNICAL *    B3-2
TECHNICAL * GROUPS - INTRODUCTORY REMARKS.= +ION OF     110
TECHNICAL * SERVICES.= +DINATION AND INTEGRATION OF     111
TECHNICAL * SERVICES.=.....CONTINUING EDUCATION IN      115
TECHNICAL * SYSTEMS IN CURRENT USE.= +CIENTIFIC AND    B3-2
TECHNOLOGY.= +EVIEW: ANNUAL REVIEW OF * SCIENCE AND    B3-2
THESAURUS AND DUAL DICTIONARY.= + COMPUTER-COMPILED     124
TRAINING PROGRAMS FOR * SCIENTISTS.= + AND ACADEMIC     118
VERIFICATION OF STRUCTURAL * .= +EM. 1. STORAGE AND     43
```

Figure 4.7 DKWIC index entries for the same high-density term of Figure 4.1 illustrating ordered access to all secondary concepts represented by significant words in the titles

Thus, the main term "INFORMATION SYSTEM" would appear in the DKWIC index gathering related subordinate terms and allowing one to quickly coordinate other concepts, as well.

There is no theoretical upper limit to the length of multi-word main terms; however, a practical limit of three or four words appears to be of sufficient magnitude to

```
INFORMATION SYSTEM
  CHEMICAL * .= + NONUNIQUE NOTATION IN A LARGE-SCALE    101
  EDITORIAL: A NATIONAL * .=.....                        E 61
  NATIONAL * .=.....EDITORIAL: A                         E 61
  NONUNIQUE NOTATION IN A LARGE-SCALE CHEMICAL * .= +    101
  NOTATION IN A LARGE-SCALE CHEMICAL * .= + NONUNIQUE    101
```

Figure 4.8 Illustration of a two-word main term which provides immediate access to more specific concepts

generate most useful multi-word concepts. Figure 4.9 illustrates how a useful three-word main term describing concepts scattered in each of the indexing schemes previously described are gathered under the term "TECHNICAL INFORMATION SERVICES".

```
TECHNICAL INFORMATION SERVICES
  CONTINUING EDUCATION IN * .=.....                    115
  COORDINATION AND INTEGRATION OF * .=.....            111
  EDUCATION IN * .=.....CONTINUING                     115
  INTEGRATION OF * .=.....COORDINATION AND             111
```

Figure 4.9 A three-word main term of a DKWIC index

The use of enrichment terms to enhance the quality of KWIC indexes applies even more so to DKWIC indexes. Two enrichment terms were added to the titles used as examples for the illustrations of this chapter - one for book reviews and one for editorials. Figure 4.6 illustrates a portion of the subordinate entries under the main term BOOK_REVIEW. Note how the subordinate entries enable one immediately to locate entries for those books whose titles contain keywords of particular interest. Furthermore, as illustrated in Figure 4.7, access can be gained through the keywords of the book titles themselves - e.g. "INFORMATION".

4.3. Stoplists for the Prototype Double-KWIC Coordinate Index

Three stoplists were used to preclude the appearance of nonsignificant main terms and subordinate terms in this prototype double-KWIC coordinate index.

The Potential Main Term Stoplist consists of low index-value words which should never appear as the first word of a main index term, but which might appear in other positions of a main term. Included on this list are such words as "activities", "announcement", "applications", "approach", "assisted", etc.; all prepositions, articles, and conjunctions; and all character strings less than three.

The Subordinate Term Stoplist consists of words which should never appear as subordinate index terms or as the final word of a multi-word main index term. Included on this list are all prepositions, articles, and conjunctions; all character strings less than three; and a few words of extraordinarily low index value, such as "some", "such", etc.

These two stoplists were invoked by the algorithms which generated the main term and subordinate term entries. Consequently, these stoplists actually prevented generation of index entries containing the stoplist words in the positions indicated above.

The Actual Main Term Stoplist, on the other hand, was invoked just prior to the output formatting stage. Its function was to eliminate redundancy caused by generation of single-word and multi-word main terms which started with a common word (see section 5.3). For example, the main terms "AMERICAN" and "AMERICAN CHEMICAL" were eliminated in favor of the more specific term "AMERICAN CHEMICAL SOCIETY" since

there was complete overlap in the titles from which they were derived. In other instances, the less specific term may have been retained if there was incomplete overlap.

4.4. Advantages and Disadvantages of the DKWIC Indexing Techniques

Some of the advantages of the double-KWIC coordinate indexing technique as compared to the KWIC indexing technique and its variants have already been cited. Briefly, they may be summarized as follows:

1) The double-KWIC technique provides a greater depth of indexing.

2) Coordinate searches can be performed more easily on double-KWIC coordinate index entries, both because of the format and because of the alphabetic ordering of the subordinate terms under the main index terms. False coordinations are unlikely, as in the PERMUTERM index (sections 3.1.4 and 3.3), because contextual relationships between the main terms and the subordinate terms are preserved in each index entry.

3) Class relationships can be expressed by use of enrichment terms. When these enrichment terms appear as main headings, the members of the class are differentiated on the basis of the subordinate index terms. Specific members of a class can also be accessed through main headings describing the specific members of the class.

4) The format of the double-KWIC coordinate index entry is more readable, because it closely resembles the format of a conventional subject index entry.

The major disadvantages of the double-KWIC coordinate indexing technique over the conventional KWIC indexing technique are the increased index size and the higher costs of index production. For example, the prototype index occupied approximately four times the space required by a comparable KWIC index. Despite such an increased size relative to the conventional KWIC index, cost-return benefits could well justify the use of DKWIC indexes in place of conventional KWIC indexes in many places.

The real value of the double-KWIC coordinate indexing technique can be appreciated when it is compared with the automated articulated subject indexing technique for generating index entries from a given set of titles or title-like phrases. The DKWIC entries approach the quality of articulated entries but because of their ease of construction, which lack extensive linguistic analysis, they could be produced at considerably reduced cost.

4.5. Prototype System Design

The examples of double-KWIC coordinate indexes displayed in Figures 4.7 through 4.9 are portions of the prototype index automatically generated by the first programming procedures developed to produce double-KWIC coordinate indexes. The system designed to create this

prototype double-KWIC coordinate index was as follows (see Figure 4.10). The first phase required generation of KWIC index records from the source titles. Since all of the words appearing in the index column of the conventional KWIC index would become candidates for potential main terms in the double-KWIC coordinate index, the main term stoplist was invoked in the KWIC index step to preclude creation of index entries for all words of low index value which were not to appear as the first word of a main index term in the DKWIC index. Potential main terms for the DKWIC index were generated in the second step by extracting individual keywords or phrases (word strings) from the index column of the KWIC index. After each potential main term was extracted, the remaining portion of the title was rotated so as to create permuted subordinate entries. The subordinate stoplist consisting primarily of articles, prepositions, and conjunctions precluded generation of subordinate entries beginning with words appearing in this stoplist.

The algorithm for generating potential main terms (PMTs) defined a word as a string of characters bounded by spaces. A PMT could consist of a single word or a set of contiguous words up to some specified upper limit. If a punctuation mark occurred at the end of any word, it was removed during creation of a potential main term. Also, word strings for which the last word of the string was on the subordinate stoplist were not generated as potential

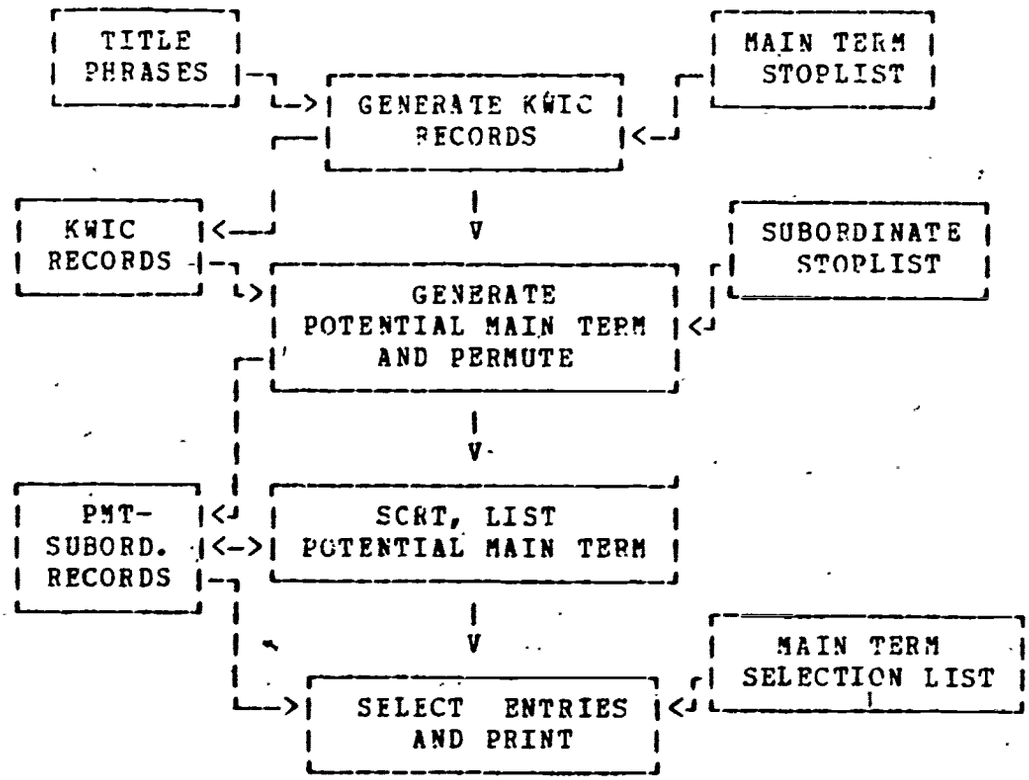


Figure 4.10 System design for creating the prototype DKWIC index

main terms.

The index records generated by the above procedures were sorted first on the basis of potential main terms and then on the basis of the words in the subordinate entry. From this sorted file, a printed list of all potential main terms generated by the procedure was obtained so that the indexer could choose the actual main terms which would appear in the final index (see section 5.3 and Figure 5.6 for further explanation of this process). These selections

were made during the final print phase via sequence numbers assigned to the potential main terms in the printed list. Selection of PMTs by sequence number rather than by stoplist (see section 4.3) proved simpler, since, on the average, fewer PMTs were selected than were rejected.

CHAPTER V. EVALUATION AND MODIFICATION OF THE PROTOTYPE SYSTEM: THE KWOC-DKWIC HYBRID INDEX

The first application of the prototype double-KWIC coordinate index algorithm provided a model to illustrate the potential advantages of this new automatic indexing technique (Petrarca,68a). Portions of this first index are displayed and discussed in Chapter 4. The construction of this and other indexes also provided opportunities for evaluation of the prototype method and suggested a number of ways in which the model could be refined. One immediately obvious refinement pertained to the often encountered situation illustrated in Figure 5.1 where the permuted subordinate terms under the main term were all derived from the same title. Obviously, there is little justification

```
LIBRARY OF CONGRESS
  APPLICATIONS IN THE * SCIENCE AND TECHNOLOGY COMPUTER    63
  COMPUTER APPLICATIONS IN THE * SCIENCE AND TECHNOLOG+    63
  DIVISION.= +LICATIONS IN THE * SCIENCE AND TECHNOLOGY.   63
  SCIENCE AND TECHNOLOGY DIVISION.= +LICATIONS IN THE *    63
  TECHNOLOGY DIVISION.= +LICATIONS IN THE * SCIENCE AND    63.
LINGUISTIC ANALYSIS
  INFORMATION RETRIEVAL.=..... LINGUIA: A * SYSTEM FOR   207
  LINGUIA: A * SYSTEM FOR INFORMATION RETRIEVAL.=.....   207
  RETRIEVAL.=..... LINGUIA: A * SYSTEM FOR INFORMATION   207
  SYSTEM FOR INFORMATION RETRIEVAL.=..... LINGUIA: A *   207
```

Figure 5.1 Size-ballooning effect in the prototype DKWIC index caused by permuting subordinate entries under main terms derived from only a single title

for ballooning the size of the index by permuting subordinate entries in situations like this. Another observation is illustrated in Figure 5.2 for those cases where an indexable word or phrase occurs more than once in a title. The title from which these particular entries were created contained the word "INDEX" twice. For each occurrence, it was extracted as if it were a different main term. Subsequent rotations of the remaining words in the title produced a stuttering effect through pairs of nearly identical subordinate entries in the resulting index. Observations such as those just described led to reexamination of the approach used to construct

---

```
INDEX
  AUTHORITY LIST TO ELIMINATE SCATTERING CAUSED BY SOM+  277
  AUTHORITY LIST TO ELIMINATE SCATTERING CAUSED BY SOM+  277
  AUTOMATICALLY GENERATED AUTHORITY LIST TO ELIMINATE +  277
  AUTOMATICALLY GENERATED AUTHORITY LIST TO ELIMINATE +  277
  COORDINATE *. II. USE OF AN AUTOMATICALLY GENERATED +  277
  COORDINATE INDEX. II. USE OF AN AUTOMATICALLY GENERA+  277
  DOUBLE KWIC COORDINATE *. II. USE OF AN AUTOMATICALL+  277
  DOUBLE KWIC COORDINATE INDEX. II. USE OF AN AUTOMATI+  277
  ELIMINATE SCATTERING CAUSED BY SOME SINGULAR AND PLU+  277
  ELIMINATE SCATTERING CAUSED BY SOME SINGULAR AND PLU+  277
  GENERATED AUTHORITY LIST TO ELIMINATE SCATTERING CAU+  277
  GENERATED AUTHORITY LIST TO ELIMINATE SCATTERING CAU+  277
  INDEX TERMS.= +AUSED BY SOME SINGULAR AND PLURAL MAIN  277
  INDEX. II. USE OF AN AUTOMATICALLY GENERATED AUTHORI+  277
  KWIC COORDINATE *. II. USE OF AN AUTOMATICALLY GENER+  277
  KWIC COORDINATE INDEX. II. USE OF AN AUTOMATICALLY G+  277
  LIST TO ELIMINATE SCATTERING CAUSED BY SOME SINGULAR+  277
```

Figure 5.2 Stuttering effect and size-ballooning effect in the prototype DKWIC index caused by permuted subordinate entries for a main term which appears more than once in a title

---

the prototype model.

The above problems obviously resulted from too close adherance to the principles of KWIC index construction. Once a potential main term was extracted from a title the remaining portion of the title was always permuted regardless of whether the potential main term occurred more than once in a given title or whether it occurred only once in the entire set of titles being indexed. Fully rotated subordinate entries were constructed for all potential main terms whether or not they were selected for inclusion in the final index. This indiscriminate approach to permuted subordinate entry construction not only created the problems mentioned before (Figure 5.1 and Figure 5.2), but also needlessly increased the cost of constructing the index. Although some of these problems had been anticipated beforehand, it was decided to generate all second order permutations of the titles for the prototype index on the premise that word and phrase patterns generated by these permutations might provide some insight into the problems of main term and subordinate term selection.

5.1. The Modified System Design: Production of KWOC-DKWIC Hybrid Indexes

To overcome many of the problems encountered in the prototype model, a slightly different approach for construction of the double-KWIC coordinate index (Lay,70) was used which produces a KWOC-DKWIC hybrid index. The

basic difference between the prototype and the modified approach are:

1) The potential main terms are now extracted directly from the titles (or title-like phrases) instead of from a KWIC index of the titles, and the potential index entries so created are temporarily retained in a KWOC-type format until other conditions are examined;

2) After all of the titles have been processed and the actual main terms have been selected, if the number of titles containing a particular main term exceeds an arbitrarily assigned threshold value, conventional double-KWIC (permuted) entries are created; if the threshold value is not exceeded, KWOC-type (non-permuted) subordinate entries are created.

The above processes are illustrated conceptually in Figure 5.3 while the system design chart for the data flow in the KWOC-DKWIC approach is illustrated in Figure 5.4. The new design consists of two phases each terminating with an alphabetic sort of records produced by that step. The first phase generates all potential main terms from the titles being indexed. The second phase is directed towards selection of actual main terms and creation of permuted subordinate entries which are to appear in the final index.

5.2. Extraction of Potential Main Terms (PMTs)

The first phase consists of the extraction of all potential main terms from the titles being indexed, word

Titles

1. COMPARING INDEXING EFFICIENCY AND CONSISTENCY.= 26
2. DOCUMENT REPRESENTATION AND INDEXING CONSISTENCY.= 56

Step 1: Potential Index Entries

DOCUMENT // * REPRESENTATION AND INDEXING CONSISTENCY.= 56
INDEXING // COMPARING * EFFICIENCY AND CONSISTENCY.= 25
INDEXING // DOCUMENT REPRESENTATION AND * CONSISTENCY.= 56
INDEXING EFFICIENCY // COMPARING * AND CONSISTENCY.= 25

1 2 7

Step 2: Actual Index Entries 3

INDEXING <
COMPARING * EFFICIENCY AND CONSISTENCY.= 25
CONSISTENCY.= COMPARING * EFFICIENCY AND 25
* CONSISTENCY.= DOCUMENT REPRESENTATION AND 56 —4
DOCUMENT REPRESENTATION AND * CONSISTENCY.= 56
* EFFICIENCY AND CONSISTENCY.= COMPARING 25
REPRESENTATION AND * CONSISTENCY.= DOCUMENT 56

DOCUMENT REPRESENTATION
* AND INDEXING CONSISTENCY.= 56 <

6 5

1 - Potential main term extracted directly from title
2 - KWOC-type subordinate entry stored temporarily with PMT
3 - Actual main term selected from potential main terms
4 - Permuted subordinate entries created from KWOC-type entries
5 - Non-permuted subordinate entry
6 - Location of the extracted main term
7 - Accession code

Figure 5.3 Annotated description of the construction of index terms for the KWOC-DKWIC hybrid index

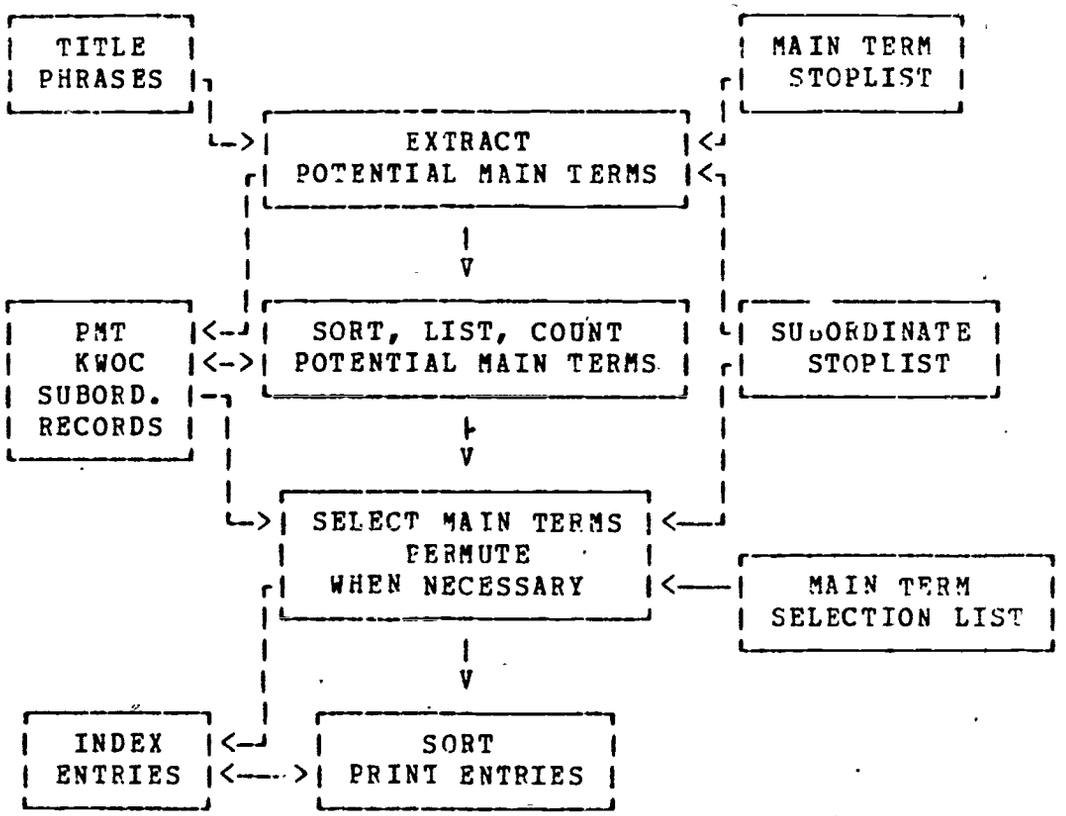


Figure 5.4 System design for creating the KWOC-DKWIC hybrid index

significance still being based on appropriate stoplists. The algorithm for generating potential main terms was modified to define a word as a string of characters bounded by a set of delimiters. These delimiters are partitioned into two classes, terminal and non-terminal, and the function of each is described below in conjunction with criteria used fo construction of PMTs. For a clearer understanding of these criteria the reader is referred to

Figure 5.5 which provides several examples illustrating their application.

A potential main term may consist of a single word or a set of contiguous words up to some specified upper limit (indicated by a user input parameter); it must have the following three attributes:

1) The first word of the potential main term must not be on the main term stoplist or on the subordinate term stoplist;

2) The last word of a candidate contiguous set must not be on the subordinate stoplist;

3) All words in a candidate contiguous set must be separated by non-terminal word delimiters.

The first and second attributes are the same as those previously required in the KWIC indexing and potential main term generation phases, respectively, of the prototype DKWIC system. Because certain punctuation marks between words usually signify introduction of a new concept, requirement of the third attribute was introduced to assure generation of potential main terms describing only a single concept. Finally the new approach identifies multiple occurrences of a potential main term in any particular title being indexed; hence, only unique potential main terms are generated from a given title.

Figure 5.5 illustrates the potential main terms that would be generated from a title on the basis of the above

Word Delimiters

Terminal	".,;:?()!"
Non-terminal	" -/"

Title

DASAR: COMPUTER-BASED DATA STORAGE AND DATA RETRIEVAL.= 62

Potential Main Terms

DASAR
COMPUTER
COMPUTER BASED
COMPUTER BASED DATA
DATA
DATA STORAGE
STORAGE
STORAGE AND DATA
RETRIEVAL

Some Potential Index Entries

DATA // DASAR: A COMPUTER-BASED * STORAGE AND * RETRIEVAL.= 62
DATA STORAGE // DASAR: A COMPUTER-BASED * AND DATA RETRIEVAL.= 62

Figure 5.5 Illustration of the effect of word delimiters and selection criteria on generation of potential main terms and potential index entries from a title. The PMTs are sequenced on the basis of the order in which they would be generated from each significant starting word in the title. The word "BASED" appeared on the primary stoplist and "AND" is on the secondary stoplist.

criteria. For one of the potential index entries illustrated therein, note the treatment for multiple occurrences of a main term "DATA" in a title. This treatment precluded the possibility of generating groups of nearly identical subordinate entries to produce the stuttering effect encountered in the initial model (Figure 5.2).

5.3. Human Interface Requirements for the Selection of Actual Main Terms (AMTs) and KWOC-DKWIC Threshold Values

After all the titles from a given source have been processed and the potential index entries have been sorted, a printed list of all potential main terms, referenced by sequence number, is prepared. This list also includes frequency data for the number of titles in which that particular main term occurs. Figure 5.6 illustrates some potential main term listings from a particular production run.

Seq#	Freq	Potential Main Term
29	13	DATA
30	1	DATA AQUISITION
31	3	DATA BASE
32	2	DATA RETRIEVAL
33	2	DATA RETRIEVAL SYSTEM
34	4	DATA STORAGE

Figure 5.6 A portion of a PMT list and occurrence frequency data used for selection of actual main terms

At this point, a human interface step is required for selection of the actual main terms which are to appear in the KWOC-DKWIC index. The sequence numbers for the desired main terms (e.g. #29 and #33 in Figure 5.6) and the threshold value for controlling the relative number of permuted and non-permuted subordinate entries are supplied as input parameters to the next processing step. The actual entries for the index are then sorted and printed in

accordance with any previously supplied display specifications (see Figure 5.7).

```
INDEX
  THE DOUBLE-KWIC COORDINATE * II.  USE OF AN
    AUTOMATICALLY GENERATED AUTHORITY LIST TO ELIMINATE
    SCATTERING CAUSED BY SOME SINGULAR AND PLURAL MAIN
    TERMS.=.....  277

INDEXING
  COMPARING * EFFICIENCY AND CONSISTENCY.=.....  324
  CONSISTENCY.=.....COMPARING * EFFICIENCY AND  324
* CONSISTENCY.=.....DOCUMENT REPRESENTATION AND  238
  DOCUMENT REPRESENTATION AND * CONSISTENCY.=.....  324
* EFFICIENCY AND CONSISTENCY.=.....COMPARING  324
  REPRESENTATION AND * CONSISTENCY.=.....DOCUMENT  238
```

Figure 5.7 Example of two types of subordinate entries found in a KWOC-DKWIC hybrid index

5.4. Other Features of the KWOC-DKWIC Hybrid System

An additional display feature for permuted subordinate entries under the new approach (Figure 5.7) enables one to more easily identify certain proximity relationships (and hence, semantic relationships) between main terms and subordinate terms. This is accomplished by displaying the replacement asterisk for the main term in the left hand margin of the subordinate entry when the main term immediately precedes the first word of the wrap-around entry.

The new systems design enables one to produce a range of index types which vary in size, quality (i.e., degree of KWOCness and DKWICness), and cost. This can be accomplished

simply by varying the threshold value which controls the relative number of permuted and non-permuted subordinate entries. For example by using a threshold value of zero one can produce an index in which all of the subordinate entries are permuted as was exemplified by the prototype index. On the other hand, by using an extraordinarily high threshold value one can produce a straight KWOC index. In between these two extremes, indexes with varying combinations of both types of entries can be generated. By using KWOC-type subordinate entries a considerable reduction in the size of the printed index results. But, as the number of KWOC-type subordinate entries under a main term is increased, the accessibility to subordinate concepts described therein is significantly reduced and the advantages of double-KWIC coordinate indexing are lost. However, if one is willing to concede that accessibility to subordinate concepts is not significantly reduced when the number of KWOC-type subordinate entries is small, one can achieve a significant reduction in the overall size and cost of the printed index by using a low threshold value for controlling the generation of permuted and non-permuted subordinate entries. For example, the size of the prototype index (section 4.4) was reduced by 40% simply by using a threshold value of one.

CHAPTER VI. VOCABULARY CONTROL FOR NATURAL LANGUAGE INDEXING

Proponents defend derivative indexing techniques not only because of the relative speed and ease of indexing large quantities of documents, but also because of the novelty and currency of the vocabulary used to construct the index entries themselves. Kennedy (Kennedy,63) claims "the use of the author's own terms - the alive currency of new ideas - rather than the considered reshapings to the indexing system may often be of great advantage." New concepts described by new words or new uses of words would rightfully find their place in the derivative indexes described earlier. Traditional indexing techniques would be forced to map these new concepts into previously established categories masking much of their usefulness. Several of the indexes discussed, notably KWIC, which contain the context about a keyword or phrase, present the user with a "suggestiveness" concerning other concepts or relations which exist in the remaining phrase. From these correlations the user may be led to other equally relevant access points in the index.

This very vocabulary freedom has also been cited as a common complaint of derivative techniques. The methods described operate on words with an equivalence relation based solely upon the character makeup of the words.

Synonyms, homonyms, eponyms, and neologisms cannot be resolved by machine without further in-depth analysis of the text presented for indexing. The machine's inability to resolve these language redundancies result in the scattering of index terms for a given topic throughout the index with the danger of possible retrieval loss by the user since he must anticipate each author's word usage.

The types of scattering occurring in derivative indexes can be classified according to the construct causing the scattering. Inflectional scattering is the result of words having the same prefix and word stem, but differing in inflectional ending. The words automate, automates, automated, automatic, automatically, and automation all refer to similar concepts yet may be scattered in the index because of terminal spelling differences. More serious problems occur in synonymal scattering, synonyms or near-synonyms which become separated in the index due to stem spelling differences.

The scattering in free vocabulary indexes can be reduced efficiently in two phases. For each access word in the index, first delete all causes of inflectional scattering, then, having retrieved the word stem, resolve any synonynal index scattering. The next two sections of this chapter present methods for reduction of index scattering.

6.1. Resolving Inflectional Scattering

The constituent words of an index descriptor are composed of an informative stem prefixed to variant character strings which merely enable this information to be expressed in grammatical form. When these stem suffixes participate as part of the collating sequence for ordering index descriptors in a printed index, inflectional scattering occurs as illustrated by some KWIC index entries from an issue of Chemical Titles containing the entries RAT and RATS separated by several pages of unrelated titles (Figure 6.1). Consequently, inflectional scattering can be resolved by identifying and eliminating grammatical endings of words participating in the index collection.

```
DE CONCENTRATION IN THE RAT.=+ FOOD INTAKE AND PLASMA FLUORI
IC MORPHOGENESIS OF THE RAT.=+ HYDRO BROMIDE, ON THE EMBRYON
 THE DIETARY RESTRICTED RAT.=+BY ISOLATED SMALL INTESTINE OF
ND TISSUE LIPIDS IN THE RAT.=+DELTA(7)-REDUCTASE, ON SERUM A
TRANSFERASE ACTIVITY IN RAT.=+ON THE PANTOTHENATE 4-PHOSPHO

                         .
                         .
                         .
ANDRO STERONE EXCRETION RATE.=  HYPERTENSION. ^DEHYDRO EPI
QUILIBRIUM CONSTANT AND RATES FOR THE REVERSIBLE REACTION
NCE OF MINIMUM STIRRING RATES IN GAS-LIQUID REACTORS.=+CURRE
XIDATION AND ACETOLYSIS RATES IN RIGID SYSTEMS.=  +BETWEEN O

                         .
                         .
                         .
 OF CONSTANT ABSORPTION RATES.=   +TRANSFER UNDER CONDITIONS
ODIUM CHLORIDE IN GRAIN RATION ON THE FREEZING POINT OF MILK
ADRENALINE SYNTHESIS IN RATS AFTER RESERPINE TREATMENT.=+OR
EN SULFATE FORMATION IN RATS AND MICE                +IN ANDROG
```

Figure 6.1 Inflectional scattering in a KWIC index

The consequences of inflectional scattering are equally apparent in the double-KWIC coordinate indexing technique. The main index terms are derived strictly on the basis of words which actually appear in the titles processed. This causes some scattering of information when two or more main terms contain the same word root but different inflectional endings. A portion of the prototype index where such scattering was observed because of the occurrence of singular and plural word forms is illustrated in Figure 6.2.

```
INFORMATION SYSTEM
  CHEMICAL * .= +A NONUNIQUE NOTATION IN A LARGE SCALE    192
  EDITORIAL: A NATIONAL * .= ..........................,  E 61
  LARGE-SCALE CHEMICAL * .= +A NONUNIQUE NOTATION IN A    192
  NATIONAL * .= ..........................EDITORIAL: A   E 61
  NONUNIQUE NOTATION IN A LARGE-SCALE CHEMICAL * .= +A    192
  USE OF A NONUNIQUE NOTATION IN A LARGE-SCALE CHEMIC+    192

INFORMATION SYSTEMS
  BOOK_REVIEW: NONCONVENTIONAL SCIENTIFIC AND TECHNIC+    B3-2
  COSTS OF * .= ..........................DETERMINING     101
  CURRENT USE.= +NTIONAL SCIENTIFIC AND TECHNICAL * IN    B3-2
  DETERMINING COSTS OF * .= ..........................    101
  NONCONVENTIONAL SCIENTIFIC AND TECHNICAL * IN CURRE+    B3-2
  SCIENTIFIC AND TECHNICAL * IN CURRENT USE.= +NTIONAL    B3-2
  TECHNICAL * IN CURRENT USE.= +NTIONAL SCIENTIFIC AND    B3-2
  USE.= +NTIONAL SCIENTIFIC AND TECHNICAL * IN CURRENT    B3-2
```

Figure 6.2 A portion of the prototype DKWIC index illustrating scattering due to the occurrence of singular and plural word forms

Inflectional scattering can be remedied by a stemming algorithm which is a computational procedure to reduce all words with the same root to a common form, usually by stripping each word of its derivational and inflectional

suffixes. A standard approach to stemming algorithms retrieves the stem of a word by removing an ending which matches a list of stored suffixes. Two main principles direct the matching of word endings: iteration and longest match.

An iterative algorithm is, as its name implies, a repeated removal of character strings affixed to a word. Lejnieks (Lejnieks,67) observed that suffixes are attached to word stems in a certain order, that is, there exists order-classes of suffixes. A match is sought with an ending in the terminal order-class (that order-class containing suffixes which are found at the end of words), the ending is removed, and the process repeated with the next order-class until no more matches are found. A strictly iterative technique may require many order-classes whose members may be difficult to ascertain.

The longest-match principle requires a single order-class. If more than one ending from this order-class matches a word suffix, the longest is removed. The principle is easily implemented by scanning the endings in order of decreasing length. Longest-match algorithms entail the generation of all possible combinations of affixes which requires a much higher storage overhead than the shorter lists of iterative approaches.

A suffix match may not always be a sufficient condition for ending removal with either algorithm. Qualitative and

quantitative context-sensitive conditions associated with a particular suffix may be necessary to limit the applicability of suffix deletion. The "context" refers, qualitatively, to the type of characters and, quantitatively, to the number of characters of the remaining stem if the ending is removed.

Tukey {Tukey,68} has proposed a context-sensitive, partially iterative, multilingual stemming algorithm whose endings are divided into four order-classes. It is structurally complex requiring distinct matching procedures for each order-class and context-sensitive case.

Salton {Salton,68b} and Lesk {Lesk,66} have described a stem and ending, longest-match, dictionary approach. The stem is sought by matching a complete entry from a stem dictionary with the first k characters of the word. The suffix, beginning with the k+1 st character must appear in an ending dictionary before the stem-ending pair is accepted. The single context-sensitive condition of stem-dictionary match can be easily handled by program, but the required dictionaries severely limit the algorithm's generality.

Lovins {Lovins,68} combines the iterative and longest-match techniques to good advantage and, with the addition of a context-sensitive recoding algorithm, cures many spelling exceptions which occur when some suffixes are attached to words.

6.1.1. Stemming and Recoding for Printed Indexes

The stemming techniques cited above are concerned with the algorithmic retrieval of word stems regardless of their form. The user of a printed index, unfamiliar with retrieval by stems, may be somewhat confused by descriptors composed of word stems. Consequently, at least for printed indexes, the stem must be recoded to form a word recognizable by the user. Words having similar stems must be similarly recoded to avoid interjecting secondary scattering.

Two possible approaches to recoding stems seem available:

1) using the stem, enter a dictionary and retrieve the preferred suffix - the reverse of Salton's technique for stemming, or,

2) the ending itself may be associated with a preferred suffix substitute.

The latter seems most appropriate because of its general applicability and lack of sizable stem dictionaries.

To attack the problems of stemming and recoding for printed indexes, a small subset of the possible inflectional endings was chosen for experimental study. Title phrases generally abound with nouns and nominal phrases. A high percentage of inflectional scattering in printed title indexes results from the occurrence of the same nominal stem in singular and plural forms. The stemming-recoding

technique to be described is presently limited to plural forms ending in "s"; however, the technique may be expanded readily to other inflectional endings.

6.1.2. Plural-Singular Stemming-Recoding Algorithm

An initial solution to inflectional scattering automatically generates singular words from plurals ending in "s" (Petrarca,68b). A word transformation routine, constructed empirically from the examination of the stemming algorithms mentioned above and a reverse English dictionary (Brown,63), acts on words ending in "s", and performs two functions: 1) decides whether the word is transformable (i.e. is a plural of a singular concept); and, 2) if the word is transformable, generates the singular form.

The algorithm identifies the transformability of a word by examining only a few characters preceding the final "s" and derives the singular either algorithmically or by consultation of an appropriate exception list. The description of the algorithm, given below, is divided into three parts, each describing the action taken based on the number of letters previously scanned. The prescription for forming the singular concept is given at each point where a transformable decision can be made.

Second to the last character is:
1) "s","u"
   the word is not transformable (e.g. stress, thesaurus, etc.)
2) "a","o"
   an exception list is examined for nontransformable

words (e.g. atlas, pathos, etc.). If the word is not found, the final "s" is dropped (e.g. spatulas, zeros, etc.).

3) "i"
if the third to the last character is "s", the word is not transformable (e.g. analysis, thesis, etc.); otherwise, the exception list mentioned in case 2 is examined for nontransformable words (e.g. this, etc.). If not present, the final "s" is dropped (e.g. martinis, etc.).

4) "'s"
the singular, non-possesive word is formed by dropping the "'s".

5) "e"
the third to the last character must be examined before a decision can be made (next section).

6) "all other letters"
an exception list is examined for nontransformable words ending in "consonant s" (e.g. physics, MEDLARS, etc.). If the word is not found, the final "s" is dropped (e.g. appears, admits, etc.).

Third to the last character is:

7) "e","u"
the singular word is formed by dropping the final "s" (e.g. trees, clues, etc.).

8) "h"
the singular is formed by dropping the final "es" (e.g. searches, etc.).

9) "v"
if the fourth to the last character is "l", the "v" is changed to "f" and the "es" is dropped to form the singular (e.g. halves, etc.); otherwise, the process is the same as in step 12.

10) "i"
an exception list containing nontransformable words ending in "ies" (e.g. series, etc.) is consulted. If the word is not found, the singular is formed by dropping the "ies" and adding "y" (e.g. activities, etc.).

11) "s"
the fourth to the last character must be examined before a decision can be made (next section).

12) "all other letters"
an exception list is consulted for irregularly formed singulars whose plurals end in "es" (e.g. indices, etc.). If the word is a member of this list, the singular is returned from an exceptions dictionary. If not present, the singular is formed by dropping the final "es" (e.g. zeroes, etc.).

The fourth to the last character is:

13) "e","y"
the singular is formed by dropping the final "es" and adding "is" (e.g. theses, analyses, etc.).

14) "s"
the word is transformable, but an exception list is examined for those plurals whose singulars are formed by dropping the final "ses" (e.g. busses, etc.). Words not on this list are transformed by dropping the final "es" (e.g. stresses, masses, etc.).

15) "all other letters"
the word is transformable, but an exception list is consulted for those words ending in "ses" for which singulars are formed by dropping the final "es" (e.g. thesauruses, choruses, etc.); otherwise, the singular is formed by dropping the final "s" (e.g. cases, uses, etc.).

The algorithm has performed well on a large number of data bases requiring exceptionally short exception lists. The lists were cumulatively gathered after processing several large title data bases. Our experience has shown that the word transformation routine coded in PL/I for an IBM 360 model 75, sucessfully singularizes all plurals ending in "s" at a rate of 50 per second when applied to a title data base containing 5% transformable plural words.

The resulting plural words and their recorded singulars can be used to gather these similar concepts under a single access point in an index by several means: 1) alter the data base being indexed by replacing all transformable plurals with their respective singulars, or 2) with a "preferred word", replace the occurrence of both the singular and plural forms of transformable plurals. The first alternative can be easily implemented as part of the word

transformation routine, altering the data base as a transformable plural is found. For generalized stem-recoding algorithms, however, this practice may lead a user astray through the omission of grammatical information. With a properly chosen "preferred word" giving some clue to the original grammatical construction, a user can generally reconstruct the appropriate suffix.

Following this second approach, the word transformation routine creates an authority list consisting of a "preferred word" for each plural-singular word pair found in the data

SINGULAR OR PLURAL	PREFERRED WORD
ACTIVITIES	ACTIVITY(IES)
ACTIVITY	ACTIVITY(IES)
AID	AID(S)
AIDS	AID(S)
ANALYSES	ANALYSIS(ES)
ANALYSIS	ANALYSIS(ES)
APPLICATION	APPLICATION(S)
APPLICATIONS	APPLICATION(S)
CHEMIST	CHEMIST(S)
CHEMISTS	CHEMIST(S)
COMPUTER	COMPUTER(S)
COMPUTERS	COMPUTER(S)
COST	COST(S)
COSTS	COST(S)
ELEMENT	ELEMENT(S)
ELEMENTS	ELEMENT(S)
ENTRIES	ENTRY(IES)
ENTRY	ENTRY(IES)
HALF	HALF(VES)
HALVES	HALF(VES)
INDEX	INDEX(ES)
INDEXES	INDEX(ES)

Figure 6.3 A portion of an automatically generated authority list produced by the plural-singular stemming-recoding algorithm

base. The "preferred word" is a non-specific entity which consists of the singular word followed by the plural ending enclosed in parentheses. Figure 6.3 depicts a portion of an authority list produced by the word transformation routine.

The authority list is utilized during index construction (see Figure 4.5 and Figure 5.1) to eliminate inflectional scattering. Each significant word in the title or phrase being examined is checked against the list of singular and plural words on the authority list. Whenever a match occurs, the actual word appearing in the context is replaced by the preferred non-specific index word located in the authority list. The grammatical information recorded in the suffix is not altered if the word appears in some functional location other than a potential main term.

```
INFORMATION SYSTEM(S)
  BOOK REVIEW: NONCONVENTIONAL SCIENTIFIC AND TECHNIC+  B3-2
  CHEMICAL * .= +A NONUNIQUE NOTATION IN A LARGE SCALE   192
  COSTS OF * .= .....DETERMINING                         101
  CURRENT USE.= +NTICNAI SCIENTIFIC AND TECHNICAL * IN   B3-2
  DETERMINING COSTS OF * .= .....                        101
  EDITORIAL: A NATIONAL * .= .....                       E 61
  LARGE-SCALE CHEMICAL * .= +A NONUNIQUE NOTATION IN A   192
  NATIONAL * .= .....EDITORIAL: A                        E 61
  NONCONVENTIONAL SCIENTIFIC AND TECHNICAL * IN CURRE+   B3-2
  NONUNIQUE NOTATION IN A LARGE-SCALE CHEMICAL * .= +A   192
  SCIENTIFIC AND TECHNICAL * IN CURRENT USE.= +NTIONAL   B3-2
  TECHNICAL * IN CURRENT USE.= +NTIONAL SCIENTIFIC AND   B3-2
  USE.= +NTIONAL SCIENTIFIC AND TECHNICAL * IN CURRENT   B3-2
  USE OF A NONUNIQUE NOTATION IN A LARGE-SCALE CHEMIC+   192
```

Figure 6.4 Reduced scattering in a DKWIC index as a result of applying an automatically generated authority list to words of main terms (compare Figure 6.2)

The results obtained using such an authority list during the creation of a double-KWIC coordinate index are illustrated in Figure 6.4 where the entries which were scattered in the prototype index (see Figure 6.2) are now merged under a single non-specific main term.

6.2. Synonymal Scattering

To the indexing specialist, the thesaurus has long been a useful device. Primarily constructed for vocabulary normalization, the thesaurus is a prescriptive indexing aid which provides a single preferred word-form for synonyms and near-synonyms, and for words occurring in various inflections if inflectional scattering has not been resolved.

Since machines are very adept at matching words, synonymal scattering is easily eliminated by automating the thesaurus lookup procedure. Artandi (Artandi,68) has outlined a well-formed procedure of automatic vocabulary normalization for book indexing. Once a keyword has been identified, it is subject to a matching operation in the thesaurus. A match signals the replacement of the original keyword with the preferred word supplied by the thesaurus.

Artandi's approach applied to natural language indexing, though normalizing the vocabulary and thus reducing synonymal scattering, reshapes the index into predetermined categories. Any connotation or suggestiveness supplied by the replaced word has been lost. A complete

change in meaning could possibly result if single words are replaced by synonyms in a title phrase. A KWIC index of such phrases would inadvertently lead a user astray.

Highcock {Highcock,68} has demonstrated the inclusion of synonymal pointers within KWIC indexes in the form of "see also" cross references. Any synonyms are included manually as part of the data base being indexed. The KWIC indexing algorithm appropriately selects all the keywords in the "see also" cross reference, placing them within the collections of like terms (see Figure 6.5).

```
  LASERS AND LASER MATERIALS. =
 AND ADVANTAGES OF MAZZONI PROCESS. = +NG OF SCAPS. OUTLINE
                HOT MELT ADHESIVES IN EUROPE. =
R. =            HOT MELT APPLIER LAYS DOWN DOT OR SPRAY PATTE
PHILICITY ON TRANS MEMBRANE EFFLUX. = + OF INCREASING NUCLEO
+ATOMY OF THE CELL MEMBRANE. THE PHYSICAL STATE OF WATER IN+
IN SEE A+ SEE ALSO MEMBRANES SEE ALSO KERATIN SEE ALSO PROTE
IN SEE A+ SEE ALSO MEMBRANES SEE ALSO KERATIN SEE ALSO PROTE
+DUCTION OF POROUS MEMBRANES FOR BATTERIES AND FUEL CELLS. =
IPIDS OF BACTERIAL MEMBRANES. =                              L
                   MERCURY SEE ALSO ELECTROCHEMISTRY. =
                   MERCURY SEE ALSO ELECTROCHEMISTRY. =
RUCTURES OF LIQUID MERCURY AND LIQUID ALUMINIUM. =           S
Y ADSORBED IONS ON MERCURY. = +SURFACE EXCESS OF SPECIFICALL
SORPTION IN SODIUM METABORATE SOLUTION. =      ULTRASONIC AB
                   METAL SEE ALSO ORGANOMETALLIC. =
                   METAL SEE ALSO ORGANOMETALLIC. =
ANEOUS TOXICITY OF METAL COMPOUNDS. =                    PERCUT
POLAR MOLECULES ON METAL OXIDE SINGLE CRYSTALS. = +RPTION OF
```

Figure 6.5 Syncnymal pointers found in a KWIC index as "see also" cross references

Automated "see also" referencing combines the two approaches mentioned above. As keywords are identified during the indexing process, matches would be recorded with

thesaurus entries. The termination of the normal keyword selection phase would signal an inspection of the thesaurus. A "see also" reference would be generated for each term whose related term also appeared in the data indexed.

"See also" cross references alert the user to synonyms present in the index but do not alter the ordering of actual index terms. The user is forced to perform this restructuring by following the synonymal pointers and examining those related entries.

----------REGENERATE----------

An electron microscopic study of REGENERATING ADRENAL gland during development of adrenal regeneration hypertension.
Nickerson PA 69 -AJPA-57-2-335

REGENERATION of HYPOTHALAMIC nerve fibers in goat
Beck E 69-NUND-5-3/4-161

Influence of nerve on lower JAW REGENERATION in adult Newt, Triturus viridescens
Finch RA 69-JOMO-129-4-401

Effect of X-irradiation on activity of protein synthetizing systems from REGENERATING RAT liver at early periods after partial hepatoectomy
Khanson KP 69-VMDK-15-6-584

Relationship of glutathione to mitotic activity in REGENERATING RAT liver
Cernoch M 69-PHBO-18-2-161

Mixed bed de-ionisatin by weak electrolyte ion-exchange RESINS REGENERATED in situ by carbon dioxide
Kadlec V 69-JACH-19-12-3 52

REGENERATION of TASTE buds after reinnervation by peripheral or central sensory fibers of vagal ganglia
Zalewski AA 69-EX NE-25-3-429

Mechanisms of REGENERATION of YEAST protoplasts. Electron microscopy of growing & regenerating protoplasts of nadsonia elongata
Havelkova M 69-FOBL-15-6-462

Figure 6.6 Vocabulary normalization in a PANDEX index collating preferred words but not altering the original text

---

An approach employed by CCM in the construction of the PANDEX index allows the index terms to be collected under a single access point. All main keywords are subject to the normalization of a thesaurus. Collation of the index entries are performed first on the normalized preferred word, which is printed as the main term, followed by the secondary term. The main and subordinate keywords are printed in boldface within the context without alteration. Consequently, synonyms appear grouped beneath a preferred word while the original text of the title phrase is preserved (see Figure 6.6).

6.3. Are Titles Sufficient?

The advent of KWIC and other computer-generated title indexes has caused much concern over the adequacy of titles as the sole source of indexing information. Titles are being utilized under the general assumption that there is a positive correlation between the title and content of the article.

Specific studies of title adequacy for particular journals or fields have produced varying results. By comparing the subject entries in Physics Abstracts with words appearing in the titles of selected articles, Maizell [Maizell,60] found that 69% of the entries for these papers were directly derived from title words. Ruhl [Ruhl,64] found that between 50% and 90% of author-prepared titles did fully reflect those index terms assigned by human indexers.

The variations observed reflected different subject fields examined, the more specific the subject area, the better the title.

Janaske (Janaske,62) has identified two distinct types of factors which contribute to the difficulties of using titles as sources of indexing information: 1) the language habits, background, interests, and idiosyncracies of the author; 2) the interests, familiarity with the subject, language habits, imagination, and idiosyncracies of the user. The witty, punning, deliberately non-informative or so called "pathological title" falls into this first category as well as the use of unfamiliar acronyms. The critical problem of bringing the user and indexer vocabulary into coincidence is the subject of the second category. Here, the searcher is forced to anticipate the terminology used by a large number of indexers (i.e. authors). Words similar only in spelling but describing different concepts or applications are grouped together. The same concepts may be expressed in quite different phraseology depending on the author's, rather than the user's, area of specialization.

Kennedy (Kennedy,63) has stressed that author participation in writing good titles is essential in this age of derivative indexing. In his suggestions to authors, he recommends:

1. consideration of the title as a one sentence abstract
2. use specific terms
3. provisions of enough context to clarify the relationships

between keywords, but no more than necessary
4. balance of brevity and descriptive accuracy
5. when possible, use words instead of notations
6. filing subjects in relation to titles to introduce general concepts in word indexes.

Herner (Herner,63) has mapped the effect of author participation from yet another, ultimately more crucial direction. He has reported a significant increase in the average number of keywords per title taken from articles appearing in the ADI and ASIS proceedings of the last decade. More recently, Tocatlian (Tocatlian,70) has suggested that the quality of titles used for articles in Chemistry has improved immeasurably since the widespread use of KWIC title indexes in Chemical Titles and other secondary publications. If these results are universal, the prognosis for titles as indexing sources is well founded.

Title enrichment offers another possibility for improving the effectiveness of titles alone. Pre-editing and augmentation of titles has been a common practice of many KWIC users. The added cost and required human analysis necessary to choose title enriching terms defeats the purpose of pure derivative indexing techniques. However, authors submitting articles to some journal publishers are required to supply pertinent "keywords" as well as an informative title and abstract. Including these enrichment terms with the title is a small price to pay for more effective retrieval.

CHAPTER VII. EVOLUTION OF THE KWIC-DKWIC HYBRID SYSTEM FOR AUTOMATING AMT SELECTION IN THE DKWIC INDEXING SYSTEMS

The index provides the primary pathway through which the researcher threading thr[illegible]ne maze of published literature retrieves his quarry. The satisfaction of success or the frustration of failure from his wanderings reflect the properties of his map, the index.

The previous chapters have described and illustrated how the double-KWIC coordinate indexing technique facilitates access to the information provided by titles at an increased level of specificity over other comparable automated indexing techniques. DKWIC, like all of these automatic indexing techniques, includes some operations which require the intervention of an index analyst. This chapter focuses attention on these human operations and discusses methods of minimizing or eliminating the need for some of them.

7.1. Magnitude of the Human Interface Requirements for The DKWIC Indexing Operations

An examination of the DKWIC construction techniques reveals three areas where an index-analyst interface is required. The first is to determine the words which have to appear on the stoplists (sections 3.2.1, 4.3, 4.5, and 5.2). The main term stoplist governs the quality of the main index terms and, to a great extent, the size of the ensuing index.

Potential main terms (PMTs) beginning with words on this stoplist are not generated, thus precluding them even from consideration in later main term selection phases. A well constructed main term stoplist enables the analyst to reject unimportant access points, and improve the overall quality of the index while reducing its size. The cost of excluding a word from the main term stoplist should exert only a minor influence on judging a word's significance. The subordinate term stoplist, too, influences the quality and size of the index. In the construction schemes previously described, the subordinate stoplist is the sole determinant of the quality of subordinate terms in permuted DKWIC index entries. In addition to prepositions, conjunctions, and articles, other words of extraordinarily low information content (e.g., some, any, etc.) should be placed on this list. By including as few as twenty-five words on this list, the number of subordinate terms generated can be reduced by as much as 40%, with a comparable reduction in the overall size of the index, and considerable improvement in the quality of both main and subordinate index terms. Such a small subordinate term stoplist is made possible by a quantitative context measurement which permits all words having less than a specified number of characters to be included as members of the list. For a new subject area, the production of stoplists can be greatly eased by the generation of a trial index to determine the vocabulary of

the data base. Once the stoplists have been created for a particular subject area they can be used repeatedly with only periodic updating.

The second operation requiring the attention of the index analyst concerns the maintenance of the singular-plural exception lists (section 6.1.2) for the vocabulary normalization procedures which have been shown to be an important tool for improving the quality of the index. These exception lists, which are required by the automatic depluralizing algorithm for eliminating inflectional scattering of main index terms, are less data dependent than stoplists but will require updating as new data bases are encountered.

The third and most critical operation requiring human intervention involves selection of the actual main terms (AMTs) which are to appear in the final index (section 5.3). These AMTs have to be selected from the PMT list generated from the particular collection of titles being indexed. The selection procedure is clouded by the subjectivity involved in determining the "worth" of a collection of main terms, a judgment weighted both by economic considerations (size of the index) and the requirement to "cover" the titles being indexed. An index is said to <u>cover</u> a collection of titles if there exists at least one actual main term (AMT) beginning with each significant word of each title of the collection. Similarly, the set of titles <u>covered</u> by a main

term is that subset of the title collection containing that main term. The remainder of this chapter deals exclusively with the problems surrounding AMT selection and culminates with a solution for automating this highly subjective manual phase of the DKWIC indexing operations.

7.2. Examination of the AMT Selection Processes

As suggested in the preceding section, an index analyst's primary concern in the AMT selection process is production of a covering index. However, he may be influenced by cost considerations to choose less appropriate actual main terms. In order to clarify this discussion of the AMT selection process and its ramifications, some notation must first be introduced.

Let A represent a potential main term and COVER(A) denote the set of titles covered by A. FIRST(A) symbolizes the first word of the phrase A.

Now, consider the following typical selection decision. Potential main term A is considered an important choice for inclusion in the final index since it singles out a significant, specific phrase common to a collection of documents. Because the index must cover the titles submitted, other main terms beginning with FIRST(A) may have to be chosen. In many cases, selections cannot be made without adding unnecessary redundancy to the final index. The potential main term B, FIRST(A) = FIRST(B), may have to be chosen to complete the covering but COVER(A) is totally

subsumed by COVER(B). Consequently, in an effort to reduce the size of the index, term B is chosen over term A even though the latter is presumably an important access point.

The method employed by the analyst in choosing these entries is facilitated by the printed potential main term statistics (Figure 5.6), which provide an indication of the size of each PMT's covering set, and the assumption that the covering sets for PMTs having the same first word and the same number of words in the PMT phrase are mutually exclusive (i.e., a single title does not contain both "INFORMATION CONTROL" and "INFORMATION SYSTEM"), an assumption which is not always valid. The sum of the covering set sizes for two-word main terms can then be compared with the size of the covering set for the corresponding single-word main term to estimate the overlap produced by selection of the two-word main terms. For high-density PMTs, this process can be extremely difficult to perform. Even with care, the selections produce considerable redundancy of entries and a proportional increase in the size and cost of the index. Furthermore, the selection problems are compounded when main term phrases having more than two words are introduced.

7.3. AMT Selection Algorithms for Minimizing Index Size and Cost

The size and cost factors influencing the selections made by the index analyst can be minimized by restructuring

the selection algorithm to allow exclusive set selection from all AMT covering sets. That is, if "INFORMATION" and "INFORMATION CONTROL" are both chosen as actual main terms, then the selection algorithm must insure that all titles containing the latter multi-word term are excluded from postings under the single-word term.

This selection burden could be passed to the index analyst by allowing him the capability to edit subordinate phrases through a selection procedure which would be executed in two steps:

1) From the PMT lists, the analyst would first choose the desired AMTs, neglecting for the time being any overlapping covering sets.

2) An AMT list with appropriate subordinate entry accession codes would be prepared, from which the analyst could eliminate those overlapping entries which were to be excluded from the final index.

Additionally, the analyst could perform finer selections at the subordinate entry level by choosing actual subordinate entries (ASEs) from each covering set of potential subordinate entries (PSEs). However, this additional task, which the analyst would have to perform manually, would make the selection processes an even greater chore than at present, particularly for large indexes.

At least the process of generating exclusive covering sets could be relegated to automatic procedures. Let us

consider the AMT selection from groups of PMTs. First, the PMTs would be segmented into mutually exclusive groups whose membership is determined by the first word of the PMT. Then, potential subordinate entries belonging to the covering set of each filial AMT in the group would be subject to set intersection with its parent AMT covering set. The actual subordinate entries associated with an AMT would include all PSEs not found in the intersections with its offsprings.

Before this approach is detailed, let us examine more carefully the structure of a PMT group. Figure 7.1 displays a typical PMT group which contains several distinct two-word and three-word potential main terms. Suppose that from this group the terms "INFORMATION", "INFORMATION PROCESSING", "INFORMATION PROCESSING CONTROL", "INFORMATION SCIENCE

Seq#	Freq	PMT
100	25	INFORMATION
101	5	INFORMATICN CONTROL
102	1	INFORMATICN CONTROL BY AUTOMATED
103	2	INFORMATION DISSEMINATION
104	1	INFORMATION DISSEMINATION TO SCIENCE
105	5	INFORMATICN PROCESSING
106	2	INFORMATICN PROCESSING CONTROL
107	1	INFORMATICN PROCESSING UTILITY
108	3	INFORMATION SCIENCE
109	3	INFORMATICN SCIENCE PROGRAMS
110	6	INFORMATION RETRIEVAL

Figure 7.1 A potential main term group consisting of all PMTs which begin with the same word (see text)

PROGRAMS", and "INFORMATION RETRIEVAL" were chosen as AMTs. These AMTs can be arranged in a dependent sequence represented by a tree structure as shown in Figure 7.2.

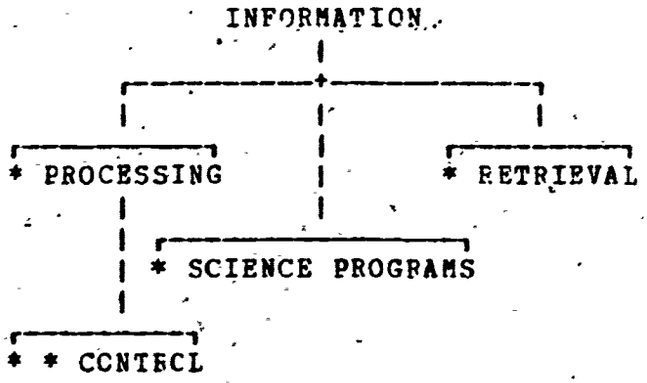


Figure 7.2 An AMT tree chosen from the PMT group of Figure 7.1

In order to discuss the relationships among elements of this tree structure, let us define some useful terminology. Let T be a directed tree with nodes {t<0>,t<1>,...t<n>}, root element t<0>, and branches {b<0>,b<1>,...b<n-1>}. A directed tree has the property that each node, except the root node, has one and only one branch directed to it. As a consequence, the branch-node relationship defines a successor function, S(t<i>), on the nodes of T such that t<j> is an element of S(t<i>) if and only if a branch of the tree is directed from node t<i> to node t<j>. The successor relationship models the dependency found in AMT trees. The successor function generates filial sets of nodes, S(t<i>) = {t<i<1>>,t<i<2>>,...,t<i<m>>}, and nodes having empty

successor sets are called terminal.

In the AMT group cited in Figure 7.2, the root of the tree is signaled by the single-word main term INFORMATION. The successors of the root element are signified by S(INFORMATION) = { * PROCESSING, * SCIENCE PROGRAMS, * RETRIEVAL}. Except for * PROCESSING, the elements of this set are terminal. The sole successor of * PROCESSING is S(* PROCESSING) = {* * CONTROL}.

Each of the AMTs chosen from a PMT group contain possibly overlapping covering sets of PSEs. An algorithm for reducing these overlapping PSE sets to mutually exclusive PSE sets can be described, employing the tree structure terminology introduced above.

1) Starting with the root element of an AMT group, form the union of all PSEs associated with each node of the successor of the root element. The exclusive PSEs of the root element are the PSEs remaining after deletion of the PSE elements in the above union from the total set of PSEs assigned to the root element. If the root-element exclusive PSE set is empty, the actual main term is not selected.

2) Let each element of the filial set, S(t), act as the root element of an AMT subtree and perform the operation defined in 1) for each of these elements.

The order in which the exclusive PSEs are selected is important. From the PSEs of the root element, all PSEs of

the root's successors must be excluded and not just the exclusive PSEs of the root's successors. The algorithm may be stated symbolically in the recursive procedure below.

```
SELECTERM(T)
   1.  Z<T> = P<T> - P<S(T)>
   2.  R = next element of S(T);   no more, return
   3.  SELECTERM(R)
```

where
P<T> designates the total PSEs assigned to node T
Z<T> designates the exclusive PSEs assigned by the function "SELECTERM" to node T
S(T) designates the set of successors to node T

The function SELECTERM operates on an entire tree and is activated by an initial call SELECTERM(ROOT) where ROOT is the root of an AMT group.

The example AMT group described by Figure 7.2 requires that the PSEs of * PROCESSING, * SCIENCE PROGRAMS, and * RETRIEVAL be collected before the exclusive PSEs of INFORMATION can be determined. To perform this implied order of operations on the PMT file, major modifications of the earlier operations would be required. Either two distinct passes over a sequential PMT file would be needed, or each PMT record would have to be directly accessible. Another significant point that must be taken into consideration is the number of set exclusions necessary to compute the function SELECTERM. Even the most sophisticated algorithms for performing set intersections (or exclusions) require extensive searching of possibly lengthly lists. Should it not be possible to carry out these searches in

primary memory, the cost of direct-access secondary storage access would probably be prohibitive. These considerations led to reexamination of the approach used for PMT generation.

7.4. Influence of the PMT Generation Process on AMT Selection Algorithms

In essence, algorithms for deleting the overlap caused by non-exclusive PSE covering sets associated with elements of an AMT group (see preceding section) would require elimination of PSEs which initially had to be created and manipulated in some earlier stage of processing. Consequently, if the selection algorithms described in the last section were to be implemented, the double costs of generating and deleting subordinate entries must be borne. Therefore, a reexamination of the methods for PMT generation (section 5.1 and 5.2) was warranted.

The manual procedures by which an index analyst chooses actual main terms of an index appear to be weighted by the number of titles covered by a particular PMT if it were chosen. The reasons for basing the choice of AMT selection on occurrence statistics is well founded. The more often a phrase (i.e., multi-word term) occurs in a corpus of documents, the more important this phrase must be. In fact, this was the reason why automatic generation of multi-word main terms seemed so attractive a possibility for increasing indexing depth. The statistical information presented to

the analyst by the PMT listings helps him to tailor the AMT group on the basis of the occurrence statistics inherent in the data itself. The statistical data would be more useful if it referred to non-overlapping covering sets.

7.4.1. A Process for Generating Exclusive PSE (Potential Subordinate Entry) Sets

A closer examination of the PMT group depicted in Figure 7.1 reveals a tree in left-list form whose nodes are PMT entries. The numeric quantities listed beside each PMT indicate the number of titles in the PMT file that contain the extracted potential main term. This number is always greater than or equal to the sum of the occurrences of each covering set member. From the statistical information accompanying the PMT group, the size of the exclusive PSE sets for each node can be easily calculated (though not manipulated as stated before). The frequency count of each terminal node is a reflection of the exclusive PSE set containing this PMT. The size of the exclusive sets of non-terminal nodes can be calculated from the function given below.

Let P<t>, t an element of T, be the size of the total PSE set and Z<t> be the exclusive PSE set associated with node t. Then, $Z\langle t\rangle = P\langle t\rangle - P\langle S(t)\rangle$

where P<S(t)> is the total PSE set of all the filial nodes of t. Figure 7.3 displays the PMT tree of Figure 7.1 in another format with values of P and Z for each node of the tree.

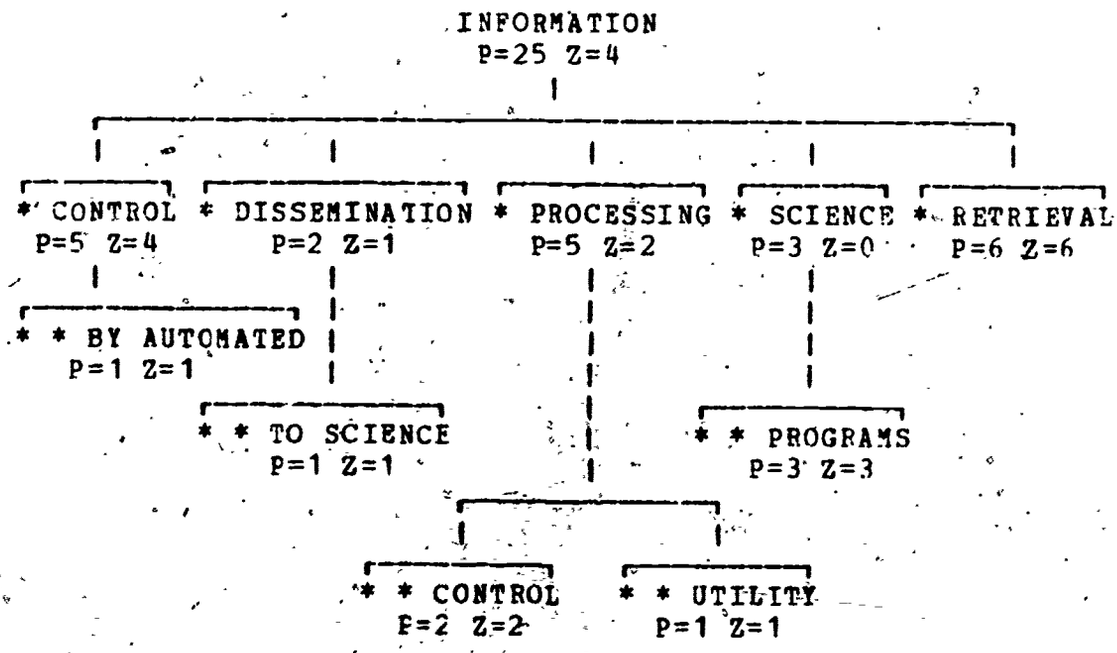


Figure 7.3 The PMT tree for the PMT group of Figure 7.1 showing values for total PSE sets (P) and exclusive PSE sets (Z) for all the nodes

The exclusivity of potential subordinate entry sets is intimately linked to the potential main terms which were extracted from the elements of these sets. The KWOC-DKWIC generation process creates these sets only for those PMTs which are terminal nodes of a PMT tree. Each non-terminal node forms a root node of a PMT subtree and the PSE set contains the terms of all successor nodes as well as terms pertaining exclusively to this node. The PMT associated with each of these exclusive sets can be distinguished during PMT generation since either the maximum size PMT (specified by a user input parameter) had been generated from this position in the title or a terminating break

character was found immediately following the PMT.

Let us assume that the PMT generation process creates only these types of entries. Can the useful PMT lists used for the prototype DKWIC model be generated? Figure 7.4 displays the terminal PMT statistics, Z<t>, that would be generated from the normal PMT list of Figure 7.1. By

Seq#	Z<t>	Terminal PMT	P<t>
100	4	INFORMATICN	25
104	4	INFORMATICN CONTROL	5
108	1	INFORMATICN CONTROL BY AUTOMATED	1
109	1	INFORMATICN DISSEMINATION	2
110	1	INFORMATICN DISSEMINATION TO SCIENCE	1
111	2	INFORMATICN PROCESSING	5
113	2	INFORMATICN PROCESSING CONTROL	2
115	1	INFORMATICN PROCESSING UTILITY	1
116	3	INFORMATICN SCIENCE PROGRAMS	3
119	6	INFORMATICN RETRIEVAL	6

Figure 7.4 Terminal PMT statistics, Z<t>, for the PMT group of Figure 7.1. P<t> represents the normal PMT statistics presented in Figure 7.1.

rearranging the expression for Z<t>, the normal PMT statistics, P<t>, can be calculated.

$$P<t> = Z<t> + P<S(t)>$$

The calculation is straightforward, though recursive. The implications for a selection algorithm, however, are not so simple. Unless all the PSEs from a given terminal PMT entry are chosen for the final index, the PSEs not chosen will have to be modified to conform to a chosen AMT. To perform the modification of both the main term and subordinate term entries, some new terminology must be introduced which

describes the generation of terminal PMTs and their PSEs. This is developed in the next section.

7.4.2. Maximal Main Terms (MMTs) and Specificity Units

To implement the processes described in the last section, a restricted set of PMTs to be generated, which are all terminal, will be called maximal main terms (MMTs). Maximal main terms are constructed from a title in segments, called specificity units. The specificity of a main term is the number of specificity units contained therein. If a maximal main term requires alteration during the AMT selection process, it is modified from one of higher specificity (i.e. having greater number of specificity units) to one of lower specificity by the deletion of specificity units from the MMT moving right to left.

Specificity units are defined formally in two classes:

1) any word not appearing on the primary stoplist;

2) the shortest contiguous string of words delimited on the left by another specificity unit and ending with a word that is not a member of the secondary stoplist.

Figure 7.5 illustrates the specificity units found in a particular title.

Combining the definition of specificity units with the previous definitions for potential main terms (section 5.2), a maximal main term has the following characteristics:

a) the first word of a MMT is a type 1 specificity unit;

Title

THE RETRIEVAL OF INFORMATION BY AUTOMATED SYSTEMS: A SURVEY

Specificity Units

Type 1
RETRIEVAL
INFORMATION
AUTOMATED
SURVEY

Type 2
OF INFORMATION
BY AUTOMATED
SYSTEMS
A SURVEY

Figure 7.5 The specificity units generated from a title. The word "SYSTEMS" appeared on the primary stoplist and the words "THE", "OF", "BY", and "A" appeared on the subordinate stoplist

b) contiguous specificity units of type 2 are contained in the MMT as long as a maximum specificity (supplied through a user input parameter) has not been surpassed, or terminating punctuation has not been found while attempting to construct the next specificity unit.

The maximal main terms that can be constructed from the title illustrated in Figure 7.5 are displayed in Figure 7.6. Typically the number of MMTs found in a title is equal to the number of significant words found therein. The specificity of each of these MMTs is dependent upon one of three factors: the input parameter indicating the maximum specificity the terminating punctuation; or the length of the title (if no terminating punctuation is used). The total

Maximal Main Term	Specificity
RETRIEVAL OF INFORMATION BY AUTOMATED	3
INFORMATION BY AUTOMATED SYSTEMS	3
AUTOMATED SYSTEMS	2
SURVEY	1

Figure 7.6 The maximal main terms formed from the specificity units illustrated in Figure 7.5

number of PMTs that could be generated from a given title is the sum of the specificities of MMTs generated from the same title. In the example above, nine PMTs would have been generated whereas only four MMTs. Assuming that a computer record of the type illustrated in Figures 5.3 and 5.5 is constructed for each PMT or MMT, then, in this example alone, less than half of the records generated for index production with PMTs would have to be generated with MMTs.

7.5 An AMT Selection Algorithm

Each MMT generated as above produces exactly one AMT in the final index such that a covering index must result. The selection procedure thus reduces to choosing the proper specificity for all AMTs from the MMTs generated. Again, we refer to the organization of the MMT groups to describe a method of manipulating these terms, and according to the definitions in section 4.2, the terminal PMT group of Figure 7.4 can now be looked upon as such an MMT group.

The MMTs can be segmented into groups in a fashion similar to the PMTs, membership being determined by the

initial specificity unit. The MMT group is agair organized as a tree in left-list form, though many intermediate nodes of the corresponding PMT tree may be absent since all elements are terminal. Note, for example, the absence of "INFORMATION SCIENCE" as an MMT in Figure 7.4 which was present as a PMT in Figure 7.1. However, all the information is present in the MMT tree to construct the PMT tree of Figure 7.3.

The actual specificity of each of the AMT selected or generated from a MMT group must be determined. Since it would be quite a chore to input that information for each entry, the following set of default AMT specificities have been designated which may be overridden by an index analyst.

1) The specificity of the first AMT of the group is 1.

2) The specificity of the next AMT of the group is the minimum of the specificity chosen for the present entry and the MMT specificity of the next MMT.

Because of the second rule, few override commands need to be applied per MMT group. In order to create the AMT specified in Figure 7.2 from the MMT group of Figure 7.4, the override commands displayed in Figure 7.7 would be necessary. Note how the remainder of the specificity tailoring would be handled by the default specificity rules.

Override Commands	Seq#	MMT
	100	INFORMATICN
	104	INFORMATICN CONTROL
	108	INFORMATICN CONTROL BY AUTOMATED
	109	INFORMATICN DISSEMINATION
	110	INFORMATION DISSEMINATION TO SCIENCE
(111,2)	111	INFORMATICN PROCESSING
(113,3)	113	INFORMATION PROCESSING CONTROL
(115,2)	115	INFORMATICN PROCESSING UTILITY
(116,3)	116	INFORMATION SCIENCE PROGRAMS
	119	INFORMATICN RETRIEVAL

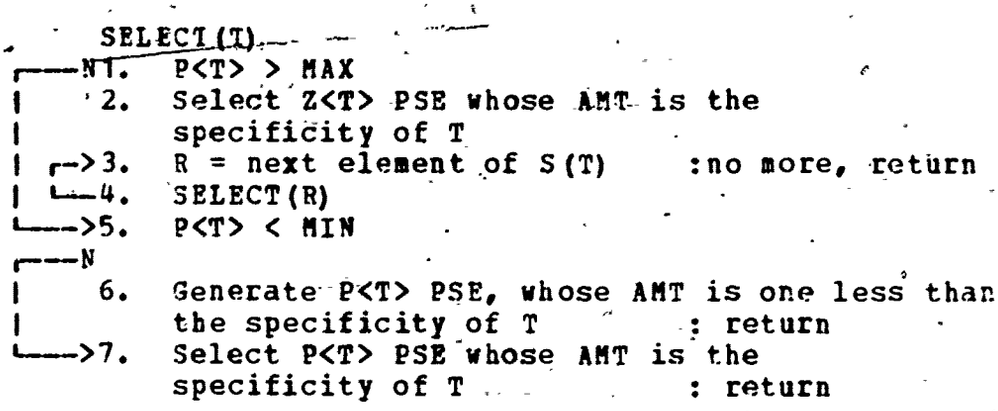
Figure 7.7 The selection override commands necessary to form the AMT selections illustrated in Figure 7.2 from the MMT group in Figure 7.4. The commands are ordered pairs of numbers signifying the sequence number of the MMT to alter and the desired AMT specificity. The underlined terms depict the AMTs selected.

7.6 Automating the AMT Selection Process

If the index analyst determines the actual main terms strictly by the frequency of occurrence of distinct concepts found in MMT groups, then the selection process itself becomes a candidate for automation {Belzer,71,Carroll,69}. Reasoning that an AMT of higher specificity is chosen over a less specific one because the less specific entry would cover too many titles, a selection algorithm can be determined.

Let us assume that an upper limit is imposed on the number of titles to be covered by an AMT. If this limit is exceeded, then AMTs will be sought at the next higher level of specificity. At this higher level, AMTs will be chosen

only if the number of titles covered by these terms meets some minimum criteria. Of course, any MMTs of lower specificity bypassed while selecting a more specific AMT will also be chosen as an AMT at the current specificity to maintain covering. The basic idea is to select AMTs covering approximately an equivalent number of titles while selecting, when possible, the most specific AMTs from the MMT group covering the titles. The algorithm, described more formally in Figure 7.8, examines the PMT tree generated from an MMT group and attempts to prune nodes so that the titles covered exclusively by each node fall between the values MIN and MAX.



where P<T> and Z<T> are, respectively, the number of total PSE and exclusive PSE of the node T, and S(T) is the set of successor nodes of T.

Figure 7.8 The logical flow for an automated main term selection process

The algorithm is called initially with the root element of a PMT tree and prunes all subtrees found therein.

Assuming that MAX is 4 and MIN is 2, the results of applying the algorithm to the PMT tree of Figure 7.3 is displayed in Figure 7.9. The actual main terms automatically selected from the PMT tree of Figure 7.3 are summarized in Figure 7.10.

```
1,1 P(INFORMATION) = 25
1,2 select 4 PSE whose AMT is INFORMATION
1,3 * CONTROL next element of S(INFORMATION)
2,1 P(* CONTROL) = 5
2,2 select 4 PSE whose AMT is INFORMATION CONTROL
2,3 * * BY AUTOMATED next element of S(* CONTROL)
3,1 P(* * BY AUTOMATED) = 1
3,6 generate 1 PSE whose AMT is INFORMATION CONTROL
2,3 no more elements in S(* CONTROL)
1,3 * DISSEMINATION next element of S(INFORMATION)
2,1 P(* DISSEMINATION) = 2
2,7 select 2 PSE whose AMT is INFORMATION DISSEMINATION
1,3 * PROCESSING next element of S(INFORMATION)
2,1 P(* PROCESSING) = 5
2,2 select 2 PSE whose AMT is INFORMATION PROCESSING
2,3 * * CONTROL next element of S(* PROCESSING)
3,1 P(* * CONTROL) = 2
3,7 select 2 PSE whose AMT is INFORMATION PROCESSING
    CONTROL
2,3 * * UTILITY next element of S(* PROCESSING)
3,1 P(* * UTILITY) = 1
3,6 generate 1 PSE whose AMT is INFORMATION PROCESSING
2,3 no more elements of S(* PROCESSING)
1,3 * SCIENCE next element of S(INFORMATION)
2,1 P(* SCIENCE) = 3
2,7 select 3 PSE whose AMT is INFORMATION SCIENCE
1,3 * RETRIEVAL next element of S(INFORMATION)
2,1 P(* RETRIEVAL) = 6
2,2 select 6 PSE whose AMT is INFORMATION RETRIEVAL
2,3 no more elements of S(* RETRIEVAL)
```

Figure 7.9 A trace of automated main term selections for the PMT tree of Figure 7.3. The numeric pairs refer to recursion level and algorithm line number respectively.

AMT	Titles covered by exclusive PSEs
INFORMATION	4
INFORMATION CONTROL	5
INFORMATION DISSEMINATION	2
INFORMATION PROCESSING	3
INFORMATION PROCESSING CONTROL	2
INFORMATION SCIENCE	3
INFORMATION RETRIEVAL	6
	25

Figure 7.10 A summary of automatic main term selections performed on the PMT tree of Figure 7.3

7.7. Automatic AMT Selection Failures and their Remedies: The KWIC-DKWIC Hybrid Index

The AMT selection algorithm discussed previously bases the selection procedure on two criteria usually used by index analysts. The first is the specificity of the potential main term since the more specific a main term, the more information conveyed to the user. The second is the number of occurrences of the PMT to determine the importance of a phrase in the context of the data base being indexed. The analyst usually chooses a more specific main term, where possible, provided there are a sufficient number of occurrences in items found in the data base.

There are situations, however, where the most specific PMT is the most appropriate even if it occurs only once in the data base (e.g., "AMERICAN CHEMICAL SOCIETY", rather than "AMERICAN", or "AMERICAN CHEMICAL", an example taken from the prototype DKWIC index). Consequently, a selection

algorithm which determines specificity of main terms solely on the basis of occurrence of phrases in the data base will fail when the technique is applied to low occurring phrases. An instance of this is shown in Figure 7.10 where the selection algorithm chose "INFORMATION SCIENCE" over the more specific term "INFORMATION SCIENCE PROGRAMS" which probably would have been chosen by an index analyst.

The occurrence frequencies of these low occurring phrases usually fall below the threshold for creating DKWIC permuted subordinate entries. Consequently, they have been formatted as KWOC-type entries in the KWOC-DKWIC hybrid index. The failure of the selection algorithm, then, results from its inability to select or create the most appropriate main term in a KWOC-type entry for these low occurring specific concepts. However, as discussed in section 3.2.1, the KWOC-type format has few advantages over the KWIC format. Extraction of main terms makes the KWOC format resemble traditional indexing formats, but the user still has to scan the context of the title to recognize fully the meaning and usage of the actual main term. The KWIC format, on the other hand, does not require the reader to search for the context about the key phrase since the remaining part of the title is immediately presented. The KWOC-DKWIC hybrid index (section 5.1) evolved as such simply because KWOC-type entries seemed to be consistent with DKWIC-type entries. However, if the index column (or key

window) of KWIC index entries were left justified, they would be equally as much compatible with DKWIC entries as are the KWOC-type entries. The KWIC-type entry would resolve the selection problem mentioned above in that the word in left-justified index column would be followed by all of the remaining words in the title, thus making the main index term for a low occurring concept as specific as needed.

```
COMPUTER(S) GRAPHIC(S)
 *  ..................................................... 085
   ANALYSIS PROGRAM: EDUCATIONAL APPLICATIONS IN ELEC+ 073
   APPLICATIONS IN ELECTRICAL ENGINEERING +DUCATIONAL 073
 * CIRCUIT ANALYSIS PROGRAM: EDUCATIONAL APPLICATIONS+ 073
   COURSE IN * ..........................AN ELECTIVE  068-3
   EDUCATIONAL APPLICATIONS IN ELECTRICAL ENGINEERING+ 073
   ELECTIVE COURSE IN * ..........................AN  068-3
   ELECTRICAL ENGINEERING +DUCATIONAL APPLICATIONS IN 073
   ENGINEERING +DUCATIONAL APPLICATIONS IN ELECTRICAL 073
   FACULTY VIEW: * ..................................  264-4
   FRESHMAN AND * ..................................THE  068-2
 * INFORMATION PROCESSES AT THE UNDERGRADUATE LEVEL ..  068
   LEVEL .INFORMATION PROCESSES AT THE UNDERGRADUATE    068
 * MEDIA ..............................................  068-1
   PROCESSES AT THE UNDERGRADUATE LEVEL .INFORMATION    068
   PROGRAM: EDUCATIONAL APPLICATIONS IN ELECTRICAL EN+ 073
   STUDENTS' VIEW: * ................................  264-2
   UNDERGRADUATE LEVEL .INFORMATION PROCESSES AT THE    068
   VIEW: * ..................................FACULTY  264-4
   VIEW: * ..................................STUDENTS'  264-2
COMPUTING FACILITIES AT THE UNIVERSITY OF ALBERTA +ME 257-3
CONSTRUCTION ENGINEERING FOR HIGH SCHOOLS +C STUDY OF 080
CONSULTATION SERVICES -- ACCREDITATION ..............  153-4
```

Figure 7.11 Display format for the KWIC-DKWIC hybrid index

Other advantages of the KWIC-DKWIC hybrid format are as follows:

1) The overall size of the index is reduced since the KWIC entries require no main term heading.

2) The size of the index is further reduced since each KWIC entry requires a single print line while the KWOC-type entries utilized in the KWOC-DKWIC hybrid index occupy as many lines as necessary to contain the entire title.

3) An accurate account of the number of lines necessary to print the index can be accumulated during the index generation process.

Figure 7.11 depicts a portion of an index in KWIC-DKWIC format.

7.8. Implementation of Automated AMT Selection in KWIC-DKWIC Hybrid Indexes

The method of constructing DKWIC index entries from maximal main terms differs considerably from either of the other DKWIC implementations previously described. With maximal main terms, no potential subordinate entries can be constructed until the specificity of the actual main term is determined. As a consequence, the generation process requires five distinct steps (Figure 7.12) which are developed in the five subsections that follow.

7.8.1. Generation of Maximal Main Terms

From the input data base, the main-term and subordinate-term stoplists, and the authority list, two files are generated. The first file is a title pointer file

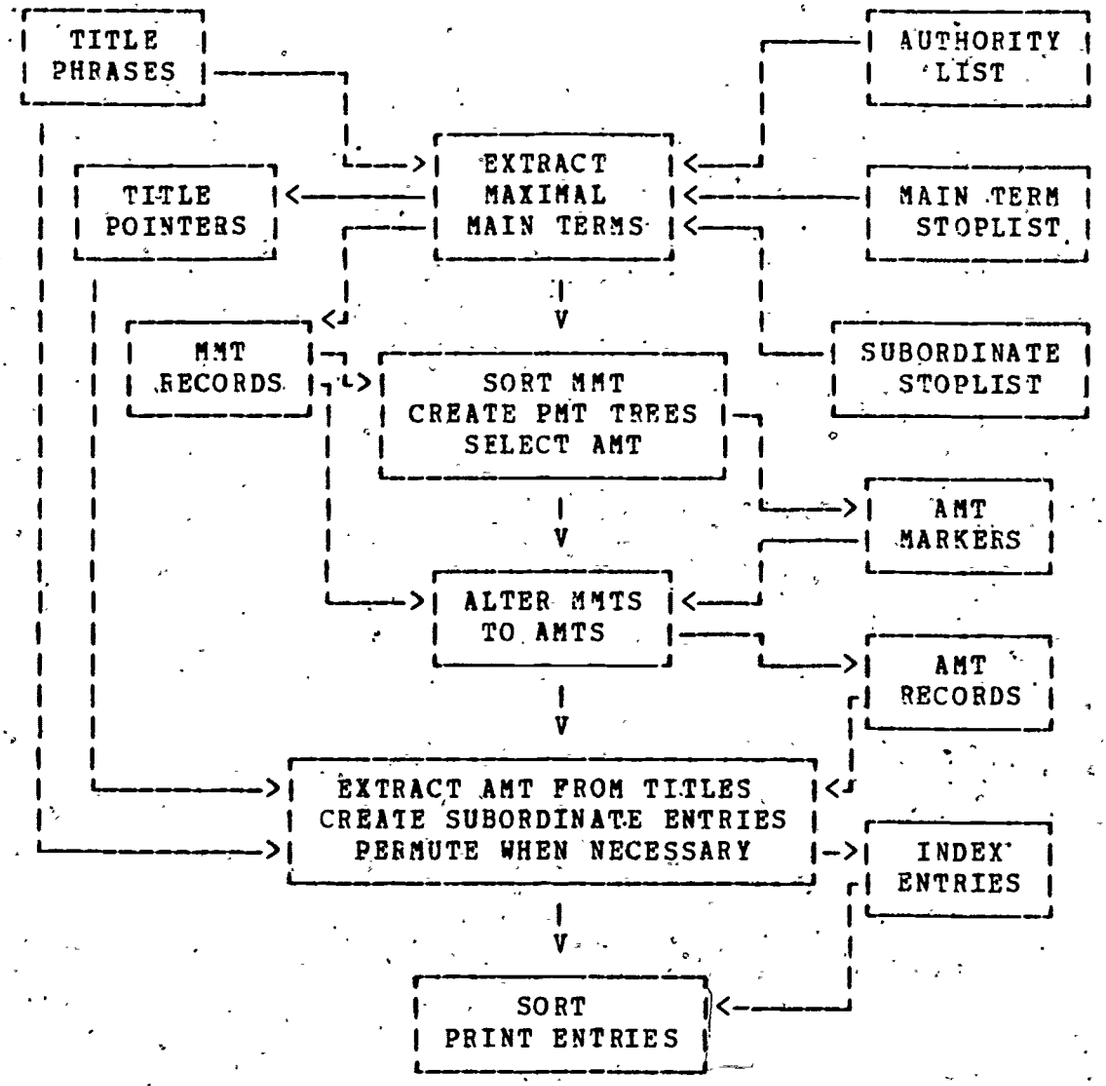


Figure 7.12 The system design for creating KWIC-DKWIC hybrid indexes with automatic AMT selection

where a fixed length record is constructed for each input title record. Each record in the title pointer file consists of five arrays which specify the location, length, main-term stoplist disposition, subordinate-term stoplist disposition, and the class of terminating punctuation for

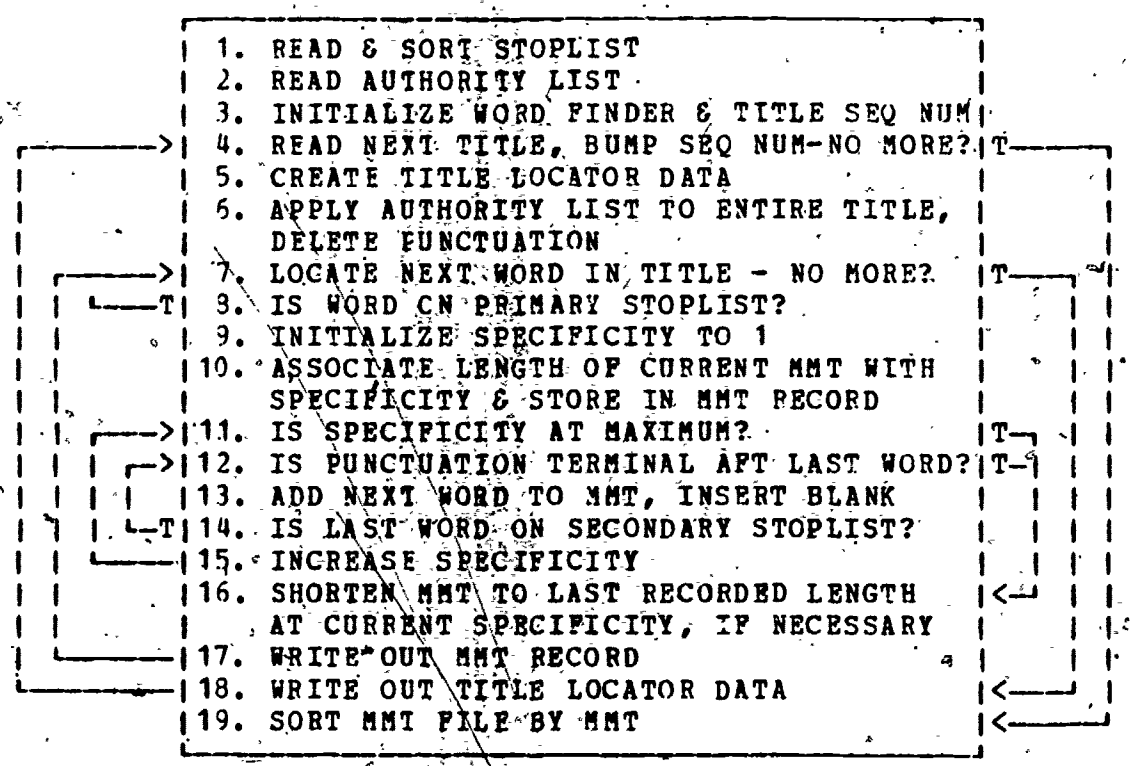


Figure 7.13 Flowchart describing maximal main term generation

each word in the title. This information is recorded at this time for later use in constructing actual subordinate entries from the corresponding title.

The second file, the MMT File, consists of all maximal main terms which could be constructed from the input title data base. Recorded with each MMT is:

a) the sequence number of the title from which it was extracted;

b) the number of specificity units found in the MMT;

c) the number of characters in any AMT generated from this MMT if a specificity less than or equal to the

constructed specificity is desired.

A simplified flowchart for generation of these files is given in Figure 7.13.

7.8.2. Selection of Actual Main Terms

The sorted MMT file acts as the prime input source for this phase of the index generation. The automated selection process consists of three distinct segments each of which is invoked for a MMT group found in the input file. The first task is to segment the MMT file into groups and, in the process, construct the PMT tree and accumulate the statistics concerning P<T> and Z<T> (see section 7.4.1) for each node of the PMT tree.

In order to conserve space, the PMT tree representation contains two entry types. The first type is a normal node entry which contains three parts: P<T> - the number of potential main terms that could be generated for this node; Z<T> - the number of terminal PMTs for this node; and a filial link to indicate the next entry in the successor set that contains this node. The second type is for terminal nodes representing PMTs of maximum specificity, where P<T> is equal to Z<T>. For these nodes only one entry in the tree structure is necessary since any brother elements will be stored consecutively in the linearized tree format. A linearized PMT tree for the MMT group shown in Figure 7.4 is illustrated in Figure 7.14. A flowchart describing construction of the PMT tree and the accumulation of the

P<T> and Z<T> statistics is depicted in Figure 7.15.

Once the PMT tree for an MMT group has been built, the AMT selection procedure outlined previously (sections 7.5 and 7.6) chooses the actual specificity of each AMT (see Figure 7.16). Since the records from the MMT file necessary to construct the PMT tree have already been processed, the selection procedure indicates the manner in which the MMTs found in the tree should be altered by creating marker

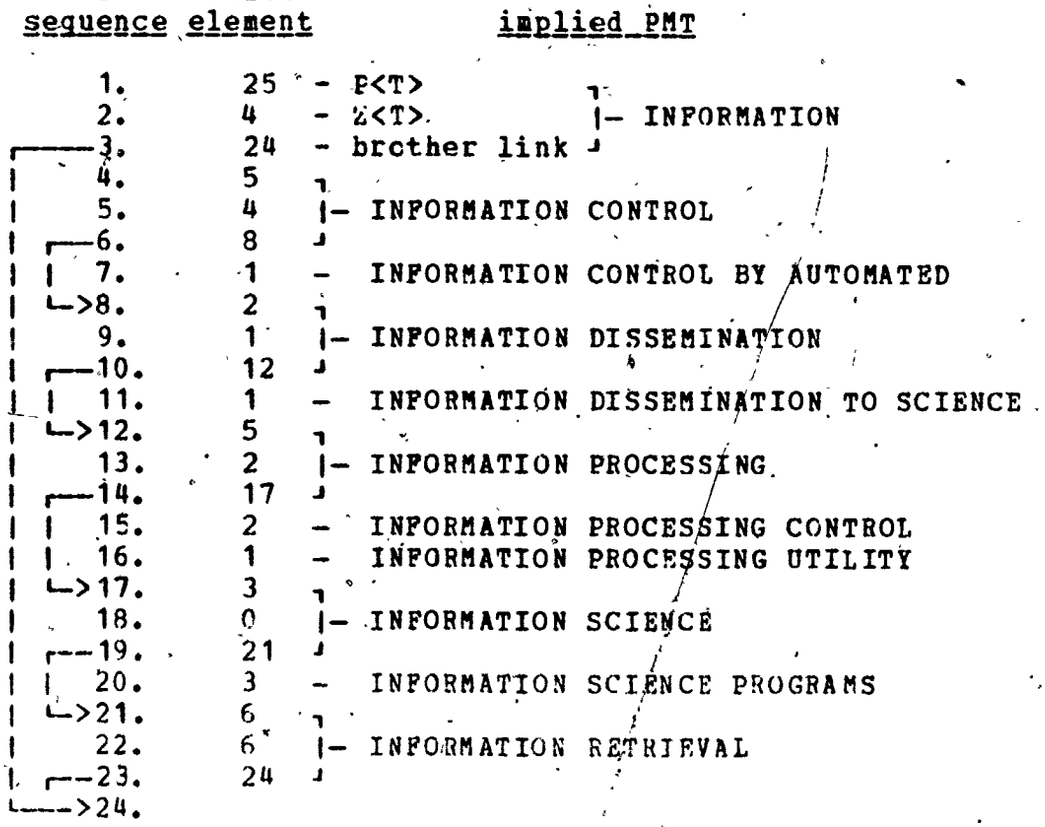


Figure 7.14 An illustration of the linearized PMT tree format for the MMT group illustrated in Figure 7.4. Only the quantities labeled "tree element" are stored.

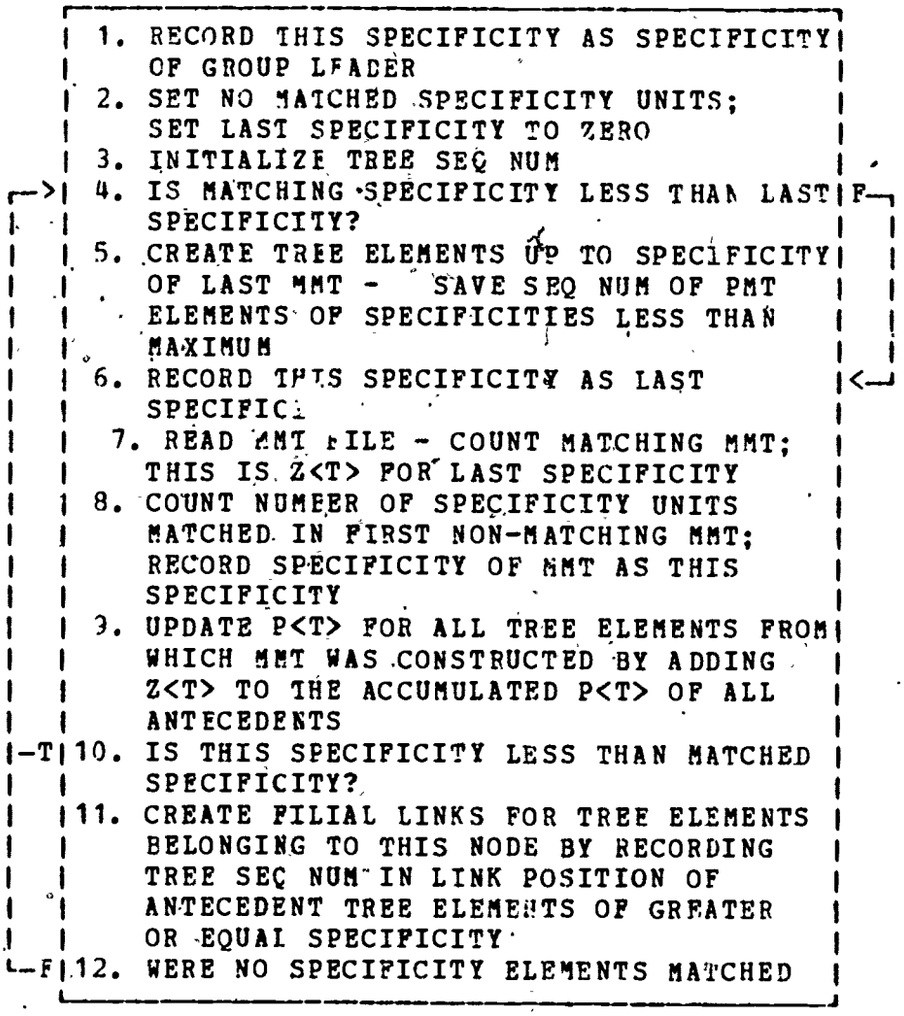


Figure 7.15 Flowchart describing the construction of a PMT tree from a MMT group

records. Each marker record consists of four items: the initial sequence number of a contiguous set of MMTs to which the selected AMT specificity applies; the ending sequence number for this set; the specificity of the AMT selected; and a fourth field, always zero, which is required for proper collation. A second type of marker is generated

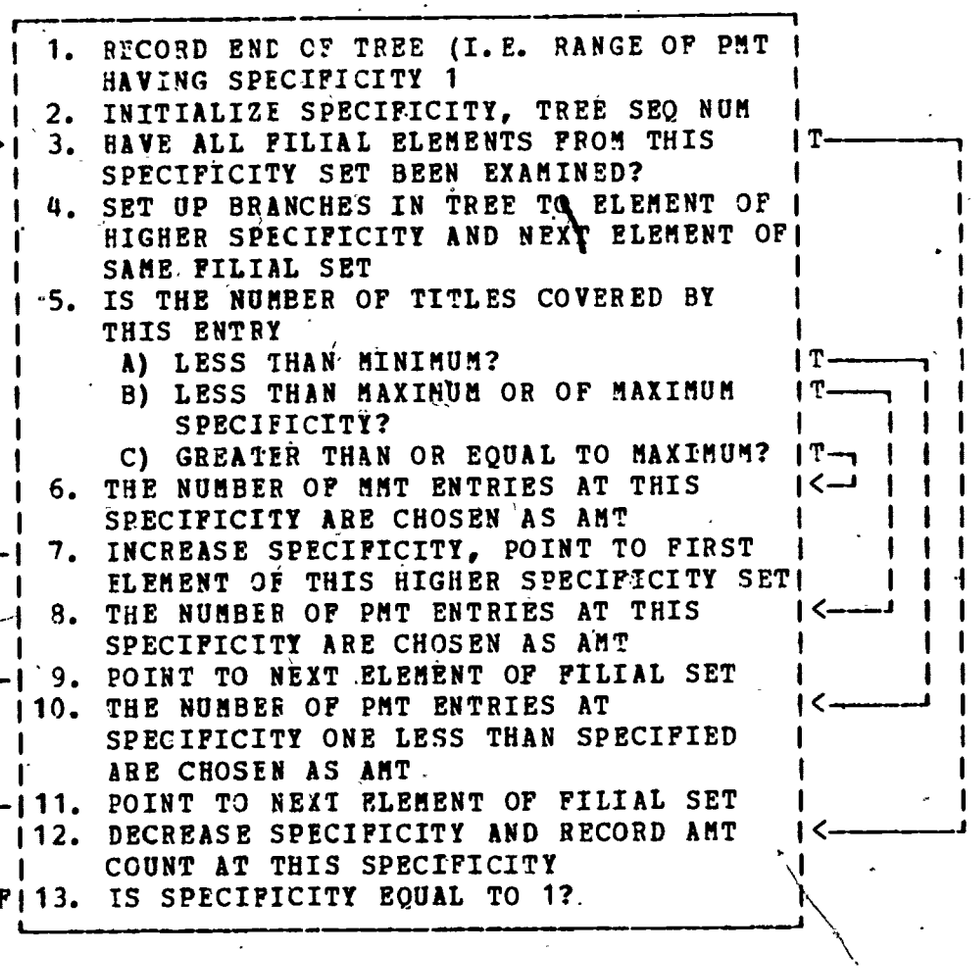


Figure 7.16 Flowchart describing the AMT selection process

which conveys the number of exclusive PSFs which a specific AMT will head. This marker is distinguished from AMT markers by a zero ending sequence number. The beginning sequence number is the MMT sequence number of the first AMT of this set. The fourth field of this record contains the exclusive PSE count. This information is placed in the

marker file to determine whether the subordinate entries of this main term should be permuted. The two marker formats are displayed in Figure 7.17.

initial MMT sequence	final MMT sequence	specificity	0

Actual Main Term Marker

initial MMT sequence	0	specificity	exclusive PSE count

Exclusive PSE Marker

Figure 7.17 The formats of the actual main term and the exclusive PSE markers produced by the AMT selection algorithm

The final step in selecting actual main terms from a MMT group involves sorting the term markers for the group. All markers are stored temporarily in main memory until all selections have been made from a group. Since the exclusive PSE markers need to be placed before all references to the MMTs they concern, the sort is performed on the first two fields of the marker records. When the sort is complete, the markers are written onto a file and the selection process continues with the next MMT group. Figure 7.18 displays a sorted set of markers and the implied selections performed on the MMT group of Figure 7.4 for maximum posting

limit of 4 and minimum posting limit of 2.

Selection Markers beg#	end#	spec	cnt	MMT
100	0	1	4	
100	104	1	0	INFORMATION
104	0	2	5	
104	108	2	0	INFORMATION CONTROL
108	109	2	0	INFORMATION CONTROL BY AUTOMATED
109	0	2	2	
109	111	2	0	INFORMATION DISSEMINATION INFORMATION DISSEMINATION TO SCIENCE
111	0	2	3	
111	113	2	0	INFORMATION PROCESSING
113	0	3	2	
113	115	3	0	INFORMATION PROCESSING CONTROL
115	116	2	0	INFORMATION PROCESSING UTILITY
116	0	2	3	
116	119	2	0	INFORMATION PROCESSING PROGRAMS
119	0	2	6	
119	125	2	0	INFORMATION RETRIEVAL

Figure 7.18 An illustration of the AMT and exclusive PSE count markers automatically produced by the AMT selection algorithm from the MMT group of Figure 7.4. A maximum posting limit of 4 and a minimum posting of 2 was used.

7.8.3. Generation of AMTs From The MMT File and AMT Marker File

The maximal main term file and the actual main term marker file are processed in parallel during this phase of the index generation (Figure 7.19). Two distinct operations are performed: the MMTs are altered to the specificity indicated by the markers produced in the last phase; and, each newly generated actual main term is coded by a field which designates the type of ASE that should be formed for this main term.

The marker file forms a non-overlapping sequence of instructions to modify each record of the MMT file. Because of the sorting technique applied during the selection phase, an exclusive PSE marker precedes the first reference to each new actual main term entry that is to be constructed (see Figure 7.18). Because of the organization of the MMT file, all maximal main terms that are to be modified to the specificity indicated by the exclusive PSE marker will be so

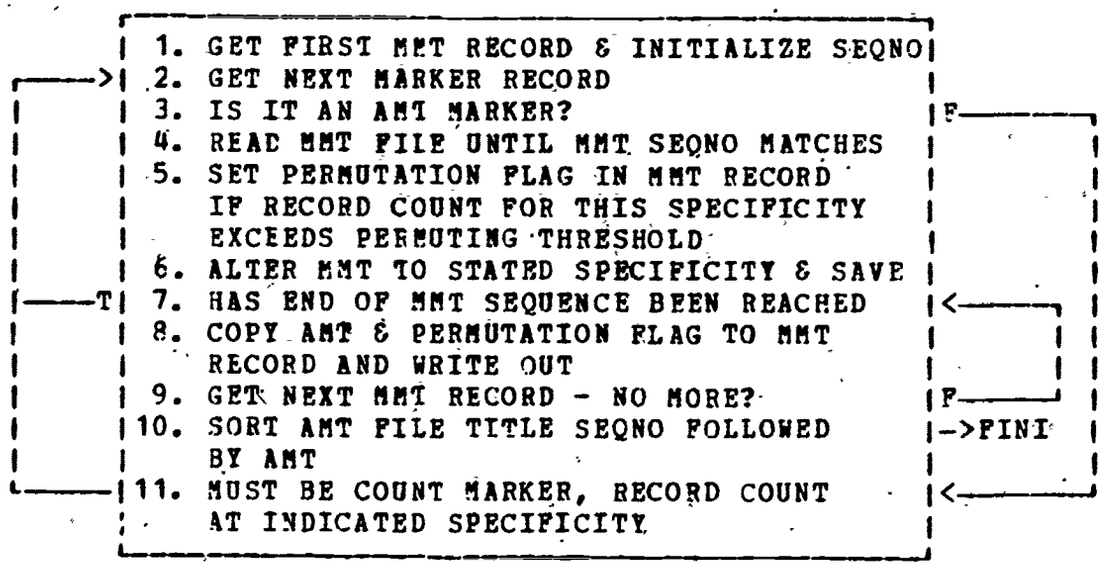


Figure 7.19 Flowchart describing the tailoring of MMT records to form actual main terms

altered before another AMT group of this specificity is encountered. An arbitrary number of AMT groups of higher specificity may appear before the termination of this AMT group. Consequently, the exclusive PSE counts are stored only by specificity.

The modified MMT records are recorded on a separate file so that the selection process may be performed again, if necessary, without requiring a re-execution of the maximal main term generation phase.

In preparation for the next step, the AMT file is sorted on the combined field of title sequence number followed by the actual main term.

7.8.4. Actual Subordinate Entry (ASE) Construction

No subordinate entries have to this point been generated, yet much information concerning them is known. The number of distinct subordinate entries is equal to the number of records in the actual main term file. A count could have easily determined how many of these terms were to form permuted subordinate entries. (In fact, by the end of the selection phase enough information can be gathered to determine an accurate estimate of the size of the index for various permutation thresholds.)

All actual main terms to be extracted from a given title are collected in an alphabetical subsequence on the AMT file prepared during the MMT tailoring phase. This arrangement allows a sequential processing of both the AMT file and the original data source. This format also permits multiple occurrences of an actual main term to be simultaneously extracted from the title and still process the AMT file sequentially (see Figure 7.20).

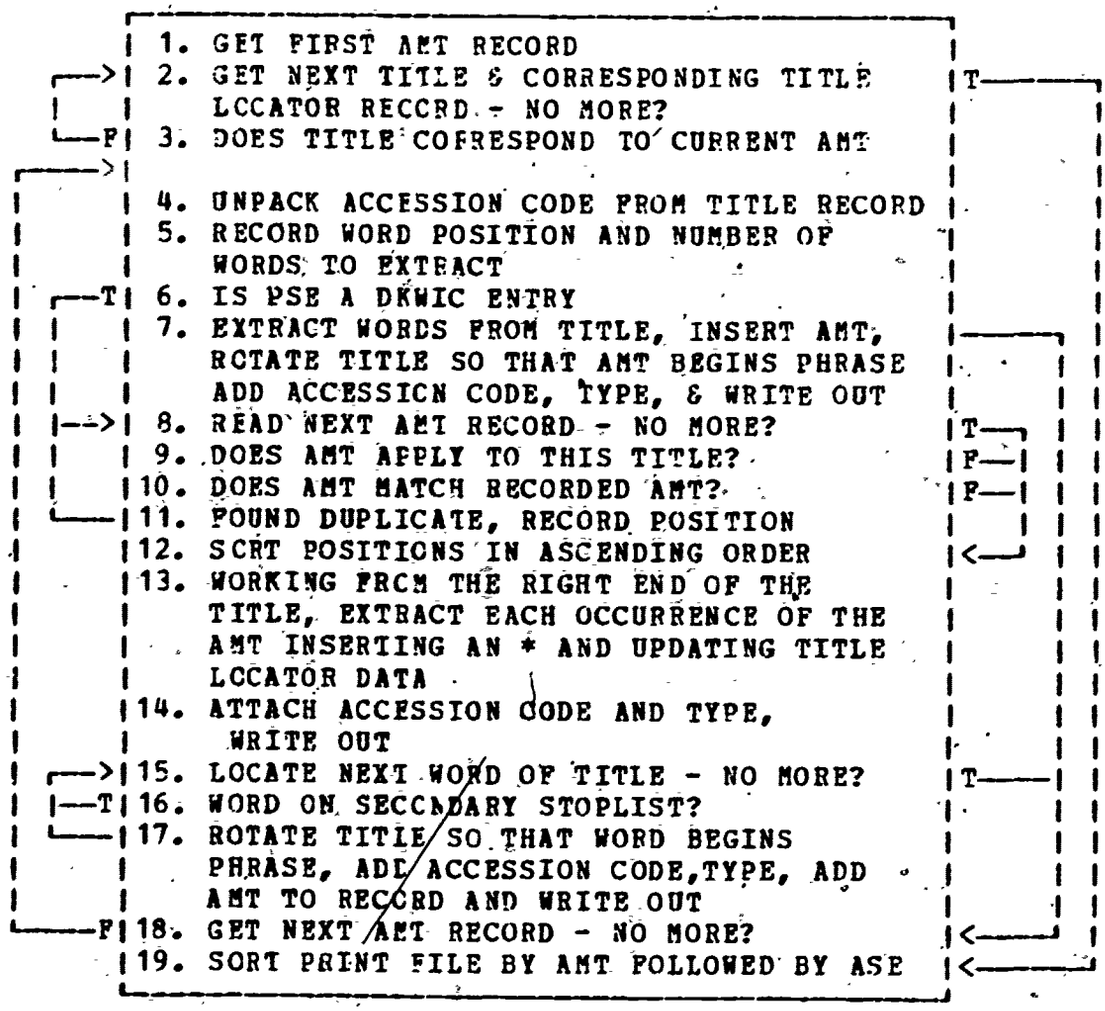


Figure 7.20 Flowchart describing the generation of ASEs

Depending upon the code set during the previous phase in each AMT record, the resulting subordinate entry is either permuted or recorded as a single entry. Subordinate index terms in permuted subordinate entries are controlled by the secondary stoplist indicator created for the corresponding title during the first phase of production.

The images recorded on the final index file contain the AMTs followed by subordinate entries and an indication of the type of formatting required.

7.8.5. Printing The KWIC-DKWIC Hybrid Index

In the final phase of KWIC-DKWIC index generation the sorted index-entry records are formatted for printing (see Figure 7.20). The width and length of a printed page are at the discretion of the user and are dynamically constructed from parametric descriptions.

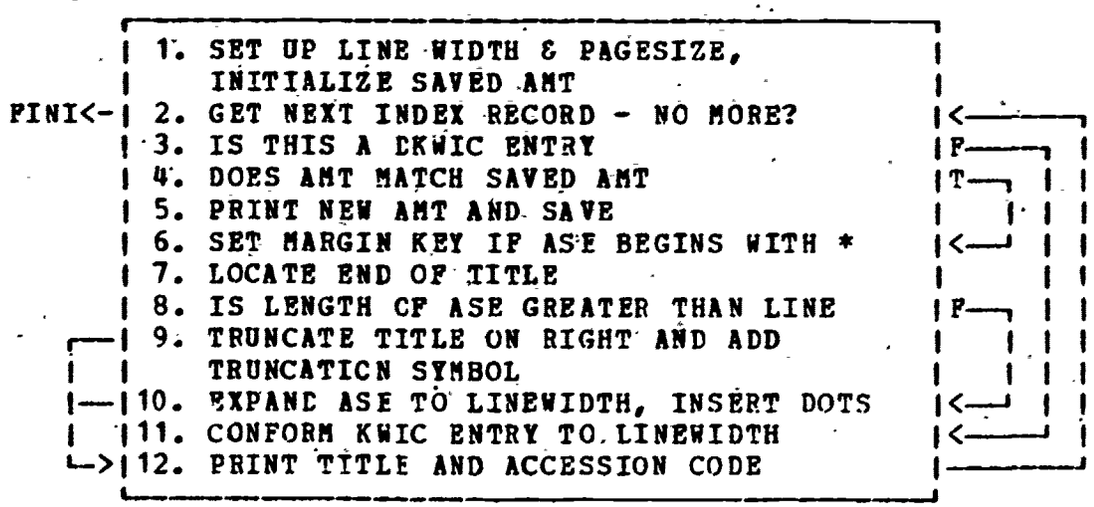


Figure 7.21 Flowchart describing the printing of the final index

CHAPTER VIII. RESULTS, CONCLUSIONS, AND DIRECTIONS FOR FUTURE RESEARCH

The capabilities of the double-KWIC coordinate indexing technique have been discussed and illustrated in previous chapters through isolated comparisons of index entries prepared by DKWIC techniques and similar entries prepared by other automated indexing schemes. In each of these examples, the DKWIC entries demonstrated properties superior to other KWIC index variants. In this chapter, I intend to demonstrate that these properties are retained in a KWIC-DKWIC hybrid index when certain selection criteria are observed. The results from this study clearly indicate roads for future improvements of the indexing system.

8.1. Influence of Various Parameters on Characteristics of the Index, and Supporting Experimental Evidence

The success of automated main term selection lies in the distribution of the words and word phrases found in the collection of titles to be indexed. This distribution is affected only by the vocabulary-normalizing functions which merge words having common stems into a single group, and the titles themselves which form the basis for the word patterns counted. The stoplists, though extremely important for determining index descriptors, dictate only which discrete groups of the word distribution should be considered in the indexing activity and which consecutive words of a title

should be chosen as main-term phrases. Consequently, the stoplist affects the content of word groups but not their distribution.

Two distinct parameters affect the specificity and the format of main terms chosen from the word-phrase distribution of terms. The posting thresholds determine which main terms should be selected from groups of terms having a common leading descriptor. The permutation threshold independently acts to divide the distribution into two groups, those main terms which will be posted with permuted DKWIC subordinate entries, and those posted as non-permuted KWIC entries.

Figure 8.1 illustrates the manner in which these two parameters affect main terms through interactions with the phrase distribution. The curve represents a rank ordering of the occurrence frequencies of distinct descriptor phrases. Experimental evidence has shown that this distribution follows Zipf's law (Zipf,49).

The posting thresholds, labeled "maximum posting" and "minimum posting" in Figure 8.1, operate locally on descriptor groups. Any member of the group which exceeds the maximum posting threshold (e.g. terms A and AB in Figure 8.1) will be altered in favor of terms which fall between the two posting limits (e.g. term ABC) while those falling below these limits are entirely eliminated (e.g. term AC). Because of the constraint of producing a covering index, the

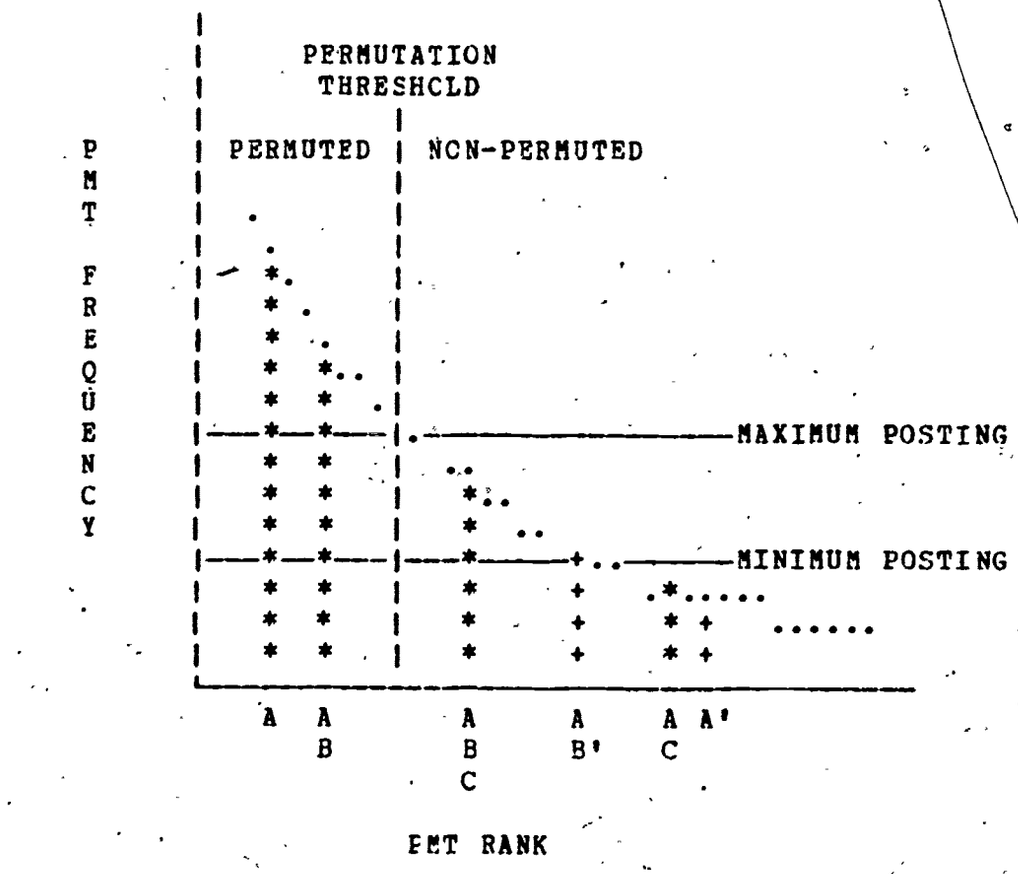


Figure 8.1 A graph illustrating influence of minimum posting threshold, maximum posting threshold, permutation threshold, and word occurrence frequency on the selection of AMTs

terms which exceed the threshold are retained in a modified form which excludes those entries covered by other terms of the group. The modified group of terms is denoted in the figure as A' and AB'. Thus, the maximum and minimum posting thresholds modify the distribution of terms as well as the selection of main terms for the final index.

The permutation threshold when applied to the distribution disregards boundaries of maximal main term groups and acts globally without concern for the decisions made by the selection process. Only the occurrence frequency is considered.

Although the permutation threshold and the posting thresholds are applied independently, their resulting interaction can affect the quality of the final index. In the example presented in Figure 8.1, the posting thresholds led to choosing the main term ABC over term AB. The resulting distribution placed the occurrence of these two terms below the permutation threshold. The terms AB and ABC would have been formatted as KWIC entries and grouped together in the index. Had the posting threshold parameters either allowed the acceptance of term AB by raising the maximum posting limit or rejected the term ABC by raising the minimum posting limit, the entries grouped under the term AB would have been selected in its original form for the final index and would have been formatted with permuted subordinate terms.

In order to further discuss these problems, some actual data from an index generation will be examined. Figure 8.2 lists the general statistics concerning the title collection. The titles of this data base were short descriptive phrases containing an average of 7.3 words per title of which an average of 2.9 words were deemed

372	titles
2702	words
391	primary stoplist words
511	secondary stoplist words
896	primary stoplist words found in titles
1627	secondary stoplist words found in titles
1075	distinct maximal main terms generated
270	specificity 1 MMTs
264	specificity 2 MMTs
541	specificity 3 MMTs
567	distinct PMT groups

Figure 8.2 Some general statistics concerning an index generation

significant after the application of the stoplists.

Table 8.1 summarizes the number of main terms selected at a particular specificity while varying the maximum and minimum posting thresholds. As was anticipated from the discussion concerning the posting threshold parameters, the average specificity of terms increased as the maximum posting threshold is decreased. This can be seen by reading either down a column in the table, fixing the minimum posting limit and decreasing the maximum, or by reading diagonally down from right to left, fixing the difference between the maximum and minimum posting threshold while each decrease by the same amount. To help clarify the interpretation of each entry, consider, for example, the quantities listed at maximum posting of 5 and minimum posting of 3. This entry indicates that at least 3 titles will be posted with each of the 13 terms at specificity 3, that at least 73 - 13 or 60 terms at specificity 2 will be

Table 8.1 A comparison of the number of main terms generated at a particular specificity as posting limits are varied.

Maximum Posting Threshold		Minimum Posting Threshold 1	2	3	4	5	6
6	# spec 1	879	981	1011	1029	1037	1047
	# spec 2	180	84	56	38	38	28
	# spec 3	16	10	8	8	0	0
	avg spec	1.20	1.10	1.07	1.05	1.04	1.03
5	# spec 1	820	959	989	1025	1037	
	# spec 2	229	101	73	37	33	
	# spec 3	25	15	13	13	5	
	avg spec	1.26	1.12	1.10	1.06	1.04	
4	# spec 1	728	944	982	1021		
	# spec 2	298	109	73	37		
	# spec 3	39	22	20	17		
	avg spec	1.35	1.14	1.11	1.07		
3	# spec 1	677	919	967			
	# spec 2	327	119	79			
	# spec 3	71	37	29			
	avg spec	1.44	1.18	1.13			
2	# spec 1	556	887				
	# spec 2	401	131				
	# spec 3	118	57				
	avg spec	1.60	1.23				
1	# spec 1	270					
	# spec 2	264					
	# spec 3	514					
	avg spec	2.35					

posted with at least 3 titles, and that at least 989 - 17 or 916 specificity 1 terms have fewer than 5 titles in common. Therefore, to insure that the higher specificity terms are not presented in the KWIC-type format in the final index,

the permutation threshold should not be greater than the minimum posting limit.

Table 8.2 illustrates the size and the fraction of DKWIC entries which were produced from the same title collection for various maximum and minimum posting limits when the permutation threshold assumes the value assigned the minimum posting limit. The size of the index increases through a maximum and then shrinks as one reads diagonally down the table from right to left. At the higher extreme of the posting limit values, the majority of the main terms have specificity one, but do not occur at sufficient

Table 8.2 Index size and the percent DKWIC-type entries of indexes prepared from the same titles with varicus posting thresholds

Maximum Posting Threshold		Minimum Posting and Permutation Threshold 1	2	3	4	5	6
6	# lines	2078	1878	1746	1567	1461	1367
	% DKWIC	76%	69%	62%	51%	43%	35%
5	# lines	1997	1854	1691	1557	1461	
	% DKWIC	73%	67%	59%	50%	45%	
4	# lines	1860	1826	1676	1557		
	% DKWIC	67%	66%	58%	50%		
3	# lines	1746	1777	1672			
	% DKWIC	61%	64%	58%			
2	# lines	1463	1700				
	% DKWIC	43%	63%				
1	# lines	1339					
	% DKWIC	40%					

frequency to surpass the permutation threshold. Thus, the majority of the entries are formatted as KWIC entries and the size of the index is small. At the lower extreme of the posting limit values, the majority of the terms have higher specificity since the maximum posting limit is small. Again, however, the majority of the entries in the index are KWIC entries since the occurrence frequency of high specificity terms is below the permutation threshold limit. I have found, through very subjective measures, that an index in which about half of the entries are permuted DKWIC entries and half are non-permuted KWIC entries appears to be the most appealing. For this hybrid index, the ideal parameters appear to be a minimum posting of 4 and a maximum limit of either 6, 5, or 4. The parametric values of 4,4, however, have the advantage of supplying the highest average specificity for the least index size.

Recall that the permutation threshold was first introduced to decrease the size of the fully permuted index. Since indiscriminant use of the permutation threshold can impair the quality of the index, further techniques must be sought to independently control the index size.

8.2. Future Research And Possible Improvements In The DKWIC Indexing Technique

Some areas of possible research and possible improvements in the DKWIC indexing technique are discussed in the next three subsections.

8.2.1. Actual Subordinate Entry Regulation

The effect of the DKWIC indexing technique on index size has been cited as one of its major disadvantages when compared with the KWIC indexing technique. The size difference results from the construction of permuted DKWIC subordinate entries. Many of these subordinate entries could lead to false coordinations with the main term because all remaining significant words in the title appear as subordinate index terms regardless of the number of distinct concepts found in a title. Reduction of the number of possible false coordinations in the index entries should improve the quality as well as reduce the size of the index produced. In some DKWIC indexes which have been produced {JCED,70, ASEE,71}, a high permutation threshold for the construction of the higher-quality DKWIC-type entries has been arbitrarily imposed because this parameter was the primary determinant of the index size after the vocabulary of the data source had been determined. Consequently, much of the power of the DKWIC format was lost because of the large number of non-permuted entries found in the index.

The reduction of the number of permuted subordinate entries generated could be used as another size-determining parameter. Furthermore, under this approach, the threshold for constructing DKWIC-type entries could be set significantly lower resulting in a higher-quality index of greater depth for a given index size.

Several approaches to limit the permuted subordinate entries appear possible. A manual subordinate entry selection procedure could be implemented, but, as pointed out earlier (section 7.2), this approach would place a considerable burden on the index analyst who would be responsible for examining each subordinate entry and choosing those having relevant coordinations with the main term. A good on-line text editing capability might alleviate much of this burden, however.

Proximity relationships between the words in the titles might afford a means of determining the more relevant coordinations algorithmically. Several approaches which would allow parameterized subordinate term selection based on distance measurements about the extracted main term are described below (see Figure 8.3 for examples).

1) Choose n significant words to the left and m significant words to the right of the extracted main term as relevent subordinate terms.

2) Delimit the boundaries of subordinate term selection by the terminal punctuation surrounding the main term.

3) Limit subordinate terms to all words up to and including the first type-one specificity unit to the left and to the right of the main term.

4) Use some combination of the three measurement criteria stated above.

Title

The Double-KWIC Coordinate Index. II. Use Of An Automatically Generated Authority List To Eliminate Scattering Caused By Some Singular And Plural Main Index Terms

Actual Main Term

AUTHORITY LIST

Subordinate Entries (only first word of subordinate entry shown)

1) choosing 2 significant words to the left and right of the actual main term

AUTOMATICALLY	ELIMINATE
GENERATED	SCATTERING

2) choosing all significant words in the interval containing the main term and bounded by terminal punctuation

AUTOMATICALLY	CAUSED
ELIMINATE	GENERATED
INDEX	MAIN
PLURAL	SCATTERING
SINGULAR	TERMS

3) choosing all significant words up to and including the next type 1 specificity unit to the left and right (underlined above) of the main term

AUTOMATICALLY
ELIMINATE
GENERATED
INDEX
SCATTERING

Figure 8.3 Subordinate terms generated by applying some word-proximity restrictions to ASE selection. The words "AN", "BY", "II", "OF" "SOME", "THE", "TO", and "USE", appear on the subordinate stoplist.

Parameterized subordinate entry selection provides an added dimension to the KWIC generation process. By varying the main term posting-permutation thresholds and the subordinate entry parameters, a wider range of indexes could be produced than could be realized by one or the other of these parameters alone.

8.2.2. Automated Generation of "See" and "See Also" Cross References

The automatic generation of "see" and "see also" cross references could result from special treatment of some stoplist entries. Consider an action which could be easily performed when a particular word is found in the stoplist. Linked to this word is a preferred index word (or phrase) which would be added as an enrichment term to the title from which the stoplist word was found. A marker indicating the presence of the stoplist word in a source title would be recorded. Processing of the enriched title would continue normally with the stoplist word not participating as a type one specificity unit. The preferred index word having been added to the title, would form a maximal main term and be chosen as an actual main term during the selection process. Each title containing the stoplist word or any other word linked to the same preferred word would be handled similarly. After all maximal main terms had been generated for the source titles, the presence markers for all special stoplist words would be interrogated. For each word that

was present in the source titles, a pseudo title would be generated containing the stoplist word, the preferred word, and "SEE" (see Figure 8.4). The stoplist disposition indicators could be set to allow indexing to occur only for the stoplist word. The normal mechanisms for generating the index would produce a main term for the stoplist word with a subordinate entry "see" reference pointing to the preferred word entry. This procedure permits title directed "see" referencing which can be a means of eliminating some scattering in the index produced by the appearance of

---

Title

MAMOS: AN IBSYS SUBSYSTEM FOR PROGRAMMING LANGUAGE EXPANSICNS.=

Index Terms

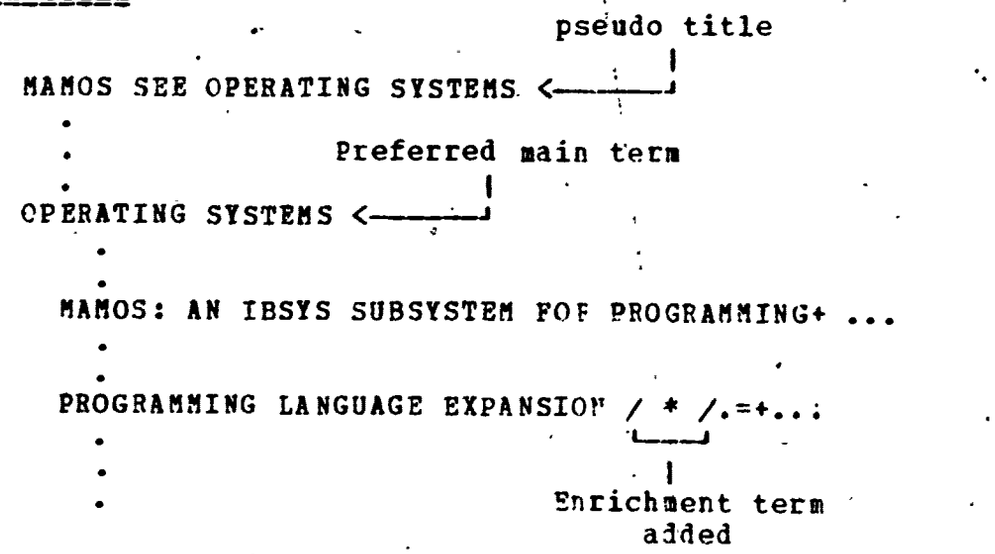


Figure 8.4 An illustration of a "see" cross reference and the enriched title from which the reference was generated

---

synonyms. With a slight modification, this procedure could automatically add enrichment terms to titles and allow the stoplist word to be indexed normally. This use would be of only minor importance if other improvements are added as explained later.

Creation of "see also" cross references for synonymally related terms could be performed in a manner similar to the creation of "see" references. The index analyst would enter related word groups which would be internally linked within the stoplist. As words are located during maximal main term generation, these related words would be marked present as they appear in titles. After the MMTs have been generated, the groups would be examined and pseudo "see also" titles generated for members of groups having two or more words marked present. The stoplist disposition of each of these words would be set so that each word would be chosen as an actual main term during later processing which would add linking "see also" records to each subordinate group.

Some "see also" cross references could be generated from statistics inherent in the main term selection process. If a significant number of high-specificity terms are selected from a PMT tree and entries for a less specific antecedent main term are also selected or generated, then "see also" cross references could be generated automatically between the antecedent and descendent main terms.

8.2.3. Other Possible Index Refining Procedures

The distance measures employed in the earlier discussions of subordinate term selection could be used in another depth increasing function. Assuming that authors construct "good" titles and the information derived from different segments of a title are interrelated, "related terms" could be automatically generated from words and phrases which lie outside the bounds of subordinate term selection. A more detailed investigation of title properties is necessary to demonstrate the feasibility of this process.

A type of scattering occurs in DKWIC indexes which is a result of multi-word main terms. This "structural scattering" is demonstrated in Figure 8.5. The main terms "INFORMATION RETRIEVAL" and "RETRIEVAL OF INFORMATION" obviously refer to the same concepts but because the indexing method treats collation differences as concept differences, scattered entries are produced. If only the significant words of a phrase were to be considered for main term generation then structural scattering would disappear. A marriage between Sharp's SLIC method (section 3.1.3) for main-term formatting and DKWIC subordinate term selection could result in a new product having the benefits of both indexing techniques. However, the deletion of actual words appearing in the title may be detrimental to the index's ability to allow valid coordinate searches. More

INFORMATION RETRIEVAL

.
.
.

RETRIEVAL OF INFORMATION

.
.
.

Figure 8.5 An example of structural scattering that occurs in double-KWIC coordinate indexes due to the syntactic structure of natural language

investigation into these properties is necessary before any conclusions can be reached.

8.3. Concluding Remarks

In conclusion, I feel that the double-KWIC coordinate indexing technique can be applied with fruitful results to existing title or title-like phrase data bases. The extensions of this new automatic indexing technique can only lead to printed indexes of higher quality requiring only minor expenditures of intellectual effort. Only through wider application and field testing of this technique and through the dissemination of its products can the real worth of these indexes be determined.

The author hopes to further improve the quality of indexes produced by these techniques and hopes to have the opportunity of continuing work along the lines mentioned previously. It is expected that several of these aspects will be investigated under continuing research performed by the Department of Computer and Information Science, The Ohio

State University.

APPENDICES

APPENDIX A. ON COUNTING ENTRIES OF AN ARTICULATED SUBJECT INDEX

Let us assume that an articulated title phrase may be stylized by letters representing components separated by function words. A phrase having four components (three articulation points) would be written as

abcd

A subject heading, extracted from the phrase, is a single component; the modifiers may be represented by a canonical notation by inserting a comma in the phrase at the point of extraction

b-a,cd

where b is a subject heading and the canonical modifier is a,cd. All subject headings and modifiers of the phrase abcd are

a-,bcd
b-a,cd
c-ab,d
d-abc,

If t<i,j> denotes the number of actual modifiers produced from a canonical form modifier having i components to the left and j components to the right of the comma, then S<n> enumerates the entries of a title phrase having n components:

$$S<n+1> = (i=0,n)\ SUM\ (t<i,n-i>)$$

In order to evaluate S<n>, some relationships among the t<i,j> must first be revealed. Define the first i components of a canonical modifier as the initial phrase and the last j components as the final phrase. Translating Lynch's rules for the construction of index entries to canonical representation:

1) if there is no initial phrase, the entry is complete;

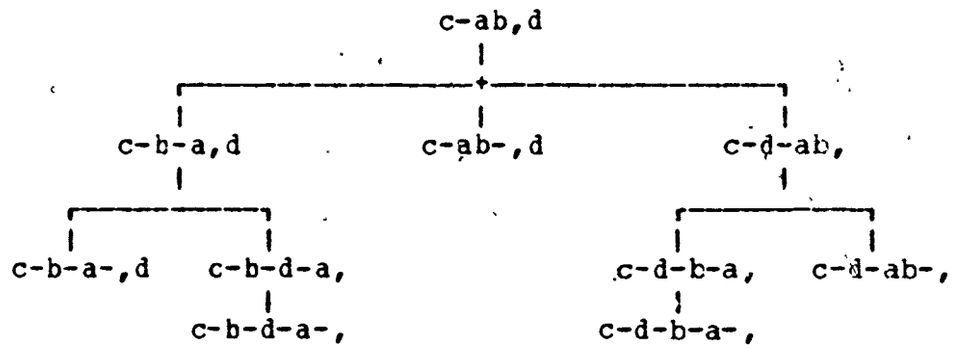
2) for each non-complete entry, subentries are formed by:

A) beginning with the last component of the initial phrase, generate i subheadings and canonical modifiers by extracting the last, the last two, ..., the last i-1, and the last i components;

B) if the initial phrase exists, extract the first and only the first component of the final phrase as a subheading.

3) continue applying 1) and 2) until all entries are complete.

The three rules given above recursively produce entries from canonical modifiers. The process may be represented by a tree structure with the terminal nodes representing the actual index entries. The tree representing the canonical decomposition of c-ab,d. is:



The terminal nodes represent the actual index entries and are punctuated as follows:

1) delete the remaining comma from the terminal form

2) replace all dashes (-) with commas except when the normal sequence of the phrase is retained (alphabetic in the example).

It is evident from the construction scheme that:

a) $t\langle 0,m\rangle = 1$ $m \geq 0$ (rule 1)

b) $t\langle n,0\rangle = (i=0,n)$ SUM $(t\langle i,0\rangle)$ $n \geq 0$ (rule 2a)

c) $t\langle i,j\rangle = t\langle i,j-1\rangle + (k=0,i)$ SUM $(t\langle k,j\rangle)$
$i,j > 0$ (rule 2a and 2b)

Applying the first difference with respect to n in b), we find

d) $t\langle n+1,0\rangle = 2t\langle n,0\rangle$

Similarly, the first difference with respect to i applied to c) yields

e) $t\langle i+1,j\rangle = t\langle i+1,j-1\rangle + t\langle i,j-1\rangle + 2t\langle i,j\rangle$

Let T(x,y) define the generating function for $t\langle i,j\rangle$

T(x,y) = (i=0,) SUM ((j=0,) SUM
(t<i,j>*(x**i)*(y**j)))

The recursion relation e) instructs the examination of

$$(y-xy+2x)\,T(x,y)$$
$$= 2x + T(x,y) - t\langle 0,0\rangle - t\langle 0,1\rangle * x$$
$$= T(x,y) - 1 + x$$

Solving for T(x,y) yeilds

$$T(x,y) = (1-x)/(1-2x+xy-y)$$

A table of some of the coefficients of the terms of T(x,y) is given in Table A.1.

Table A.1 The number of index entries generated from a title having n initial phrases and m final phrases

Final Phrase \ Initial Phrase	0	1	2	3	4	5	6	7
0	1	1	2	4	8	16	32	64
1	1	2	5	12	28	64	144	320
2	1	3	9	25	66	168	416	1008
3	1	4	14	44	129	360	968	2528
4	1	5	20	70	225	681	1970	5500
5	1	6	27	104	363	1182	3653	10836
6	1	7	35	147	553	1925	6321	19825
7	1	8	44	200	806	2984	10364	34232

Recalling that the total number of entries for a phrase,

$$S\langle n+1\rangle = (i=0,n)\ \text{SUM}\ (t\langle i,n-i\rangle)$$

is represented as the sum of the diagonals of the matrix

above. This sum can be expressed in closed form by rearranging some of the previous expressions as

$$2t<i,j> - t<i,j-1> = t<i+1,j> - t<i+1,j-1>$$

and examining

$$2S<n+1> - S<n> = 2*(i=0,n)\ SUM\ (t<i,n-i>)$$

substituting

$$- (i=0,n-1)\ SUM\ (t<i,n-1-i>)$$

$$= 2t<n,0> + (i=0,n-1)\ SUM\ (2t<i,n-i> - t<i,n-1-i>)$$

and upon substitution of the recursion relation

$$= 2t<n,0> + (i=0,n-1)\ SUM\ (t<i+1,n-1> - t<i+1,n-1-i>)$$

upon rearranging

$$= S<n+2> - S<n+1> + 2t<n,0> - t<n+1,0> + t<0,n> + t<0,n+1>$$

Substituting a) and d), all terms involving t cancel. Thus,

$$S<n+2> - 3S<n+1> + S<n> = 0$$

which can be easily solved.

Some values for S<n> are listed below:

n	S<n>
1	1
2	2
3	5
4	13
5	34
6	89
7	233
8	610

Examining the recursion relation for a Fibonacci series

$$F<i+2> = F<i+1> + F<i>$$

it is interesting to note that

$$-F\langle i+2\rangle + F\langle i+1\rangle + F\langle i\rangle + F\langle i+3\rangle - F\langle i+2\rangle - F\langle i+1\rangle + F\langle i+4\rangle - F\langle i+3\rangle - F\langle i+2\rangle = 0$$

and may be rewritten as

$$F\langle i+4\rangle - 3F\langle i+2\rangle + F\langle i\rangle = 0$$

Let i = 2n and the equation above represents S<n>, or S<n> = F<2n>. Since S<0> is undefined and S<1> = 1, S<n> actually is represented by F<2n-1>, F<0> = 0 and F<1> = 1. Consequently, S<n> is represented by the odd elements of the natural Fibonacci sequence.

APPENDIX B. ON ESTIMATING THE NUMBER OF ENTRIES OF A KWIC-DKWIC INDEX

Because of the nature of DKWIC indexing principles, the number of entries generated from a single title cannot be estimated easily from a stylized model. Many global characteristics which depend on the document collection contribute to the number of entries generated from a single title. For example, permuted subordinate entries are generated only when the number of entries to be posted beneath an actual main term exceeds a predefined threshold. Although these attributes could be estimated through probablistic analysis, the distributions required are difficult to obtain in full generality and depend heavily on the titles being indexed.

In lieu of these difficulties, the necessary distributions are calculated as part of phase 2 of the automatic selection process for generating DKWIC indexes. When an exclusive PSE frequency marker is generated by the auto-selection algorithm, the frequency is used to locate a counter in an array of counters and increment its value. After the selection process has operated on all MMT groups, the resulting array represents the density of titles collected by actual main terms.

APPENDIX C. SYSTEM INSTALLATION AND EXECUTION INSTRUCTIONS FOR THE DOUBLE-KWIC COORDINATE INDEX SUBSYSTEMS

C.1 Form Of The Distributed Indexing Subsystems

Two complete double-KWIC coordinate index subsystems consisting of 14 data sets are distributed on 9-track, OS-standard-labeled, 800 bpi tape with VOLume label DKWIC. Both the KWOC-DKWIC and KWIC-DKWIC generators are included as well as the supporting authority list generator and a model data base interface subroutine. The first 10 data sets contain the PL/I Version 5.2 source and OS/360 assembly source for the indexing systems. The object and load modules for the source programs are contained in unloaded PDSs of files 11 and 12 respectively. File 13 contains some useful JCL procedures which will aid the installation and execution of the indexing systems. The last file is a copy of this thesis in upper-lower case print form. The characteristics of these data sets are described below.

	name	format	content
1.	DKWIC.L1	FB	KWOC DKWIC source (PL/I)
2.	DKWIC.L2	FB	Chemical Titles data base interface subroutine source (PL/I)
3.	DKWIC.L3	FB	word finder subroutine source (360/BAL)
4.	DKWIC.L4	FB	authority list generator source (PL/I)
5.	DKWIC.L5	FB	KWIC DKWIC monitor source (360/BAL)
6.	DKWIC.L6	FB	phase 1 KWIC DKWIC - maximal main

term generator source (PL/I)

7. DKWIC.L7 FB phase 2 KWIC DKWIC - actual main term select source (PL/I)

8. DKWIC.L8 FB phase 3 KWIC DKWIC - actual main term modifier source (PL/I)

9. DKWIC.L9 FB phase 4 KWIC DKWIC - actual subordinate term generator source (PL/I)

10. DKWIC.L10 FB phase 5 KWIC DKWIC index print source (PL/I)

11. DKWIC.L11 IEHMOVE unloaded PDS of the 10 object modules of the programs listed above. The unloaded PDS name is DKWIC.OBJECT DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200) . The partitions are named DKWIC1 through DKWIC10.

12. DKWIC.L12 IEHMOVE unloaded PDS of the load modules for the indexing subsystems. The unloaded PDS name is DKWIC.INDEXLIB; DCB=(RECFM=U,BLKSIZE=3400). When loaded by IEHMOVE this data set can be used as a STEPLIB for index generation. The KWOC DKWIC generator is named KWODKWIC, the KWIC DKWIC generator is named KWIDKWIC, and the authority list generator is named AUTHLIST.

13. DKWIC.L13 FB sample JCL for loading, compiling, linking, and executing the DKWIC subsystems

14. DKWIC.L14 FB a copy of the print-line images of this thesis in upper-lower case. This data set should be printed with a standard TN print train. DCB=(RECFM=FB,LRECL=133,BLKSIZE=3458)

All source modules have characteristics

DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)

and can be updated with the IEBUPDTE utility.

C.2 Job Control Installation And Execution Aids

With the exception of some added descriptive comments, this section is a copy of data set DKWIC.L13. This data set should be punched and used as an aid in installing the DKWIC indexing subsystems. To punch this data set, the following model may be used:

```
// ...      JOB
//PCH       EXEC PGM=IEBGENER,
//SYSPRINT  DD   SYSOUT=A
//SYSUT1    DD   DSN=DKWIC.L13,UNIT=2400,DISP=OLD,
//           LABEL=13,VOL=(,RETAIN,SER=DKWIC)
//SYSUT2    DD   SYSOUT=B,DCB=BLKSIZE=80
//SYSIN     DD   DUMMY
```

The data set DKWIC.L13 contains job control language procedures which are placed within a job stream or optionally put in SYS1.PROCLIB. Several parameters are provided to tailor the procedures to a particular installation as noted below:

UNIT - a direct access class such as 2311 or 2314.
Default UNIT=2314.

LABEL - the label number of the data set on the distribution tape. Must be supplied where indicated.

SER - a VOLume serial number of a direct access volume on which the object or load modules are to reside. Must be supplied where indicated.

To compile a PL/I source DKWIC program:

```
//DKWICOMP PROC
//CMP      EXEC PGM=IEMAA,PARM='ATR,NEST,XREF'
//SYSPRINT DD   SYSOUT=A
//SYSLIN   DD   UNIT=SYSDA,SPACE=(TRK,(5,2)),
//         DISP=(NEW,PASS),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN    DD   DSN=DKWIC.L&LABEL,UNIT=2400,
//         DISP=OLD,LABEL=&LABEL,VOL=(,RETAIN,SER=DKWIC)
//         PEND
```

DKWICOMP compiles one of the PL/I source programs from the distribution tape and places the object program on a direct access device. This data set can be referenced by DSN=*.stepname.CMP.SYSLIN. The program compiled depends upon the LABEL parameter which must be supplied when the procedure is called.

To assemble a 360/BAL source DKWIC program:

```
//DKWICASM PROC
//CMP      EXEC PGM=IEUASM,PARM='NODECK,LOAD,XREF'
//SYSPRINT DD   SYSOUT=A
//SYSLIB   DD   DSN=SYS1.MACLIB,DISP=SHR
//SYSGO    DD   UNIT=SYSDA,SPACE=(TRK,(5,2)),
//         DISP=(NEW,PASS),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,1)
//SYSUT2   DD   UNIT=SYSDA,SPACE=(CYL,1)
//SYSUT3   DD   UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN    DD   DSN=DKWIC.L&LABEL,UNIT=2400,
//         DISP=OLD,LABEL=&LABEL,VOL=(,RETAIN,SER=DKWIC)
//         PEND
```

DKWICASM assembles one of the 360/BAL source programs from the distribution tape and places the object program on

a direct access device. This data set may be referenced by DSN=*.stepname.CMP.SYSGO. The program assembled depends upon the LABEL parameter which must be supplied when the procedure is called.

To load the object or load modules of the DKWIC subsystems:

```
//DKWICLD  PROC UNIT=2314
//LOAD     EXEC PGM=IEHMOVE
//SYSPRINT DD   SYSOUT=A
//DD1      DD   UNIT=&UNIT,DISP=OLD,VOL=SER=&SER
//DD2      DD   UNIT=2400,DISP=OLD,VOL=(,RETAIN,SER=DKWIC),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSUT1   DD   UNIT=&UNIT,DISP=OLD,VOL=SER=&SER
//         PEND
```

DKWICLD is a procedure skeleton which can be employed to load the partitioned data sets containing either the object or load modules to direct access storage. The SER parameter is required and must specify the volume name of a direct access volume. The UNIT parameter may be overridden to supply the correct direct access storage class. A LOAD.SYSIN dd statement must be supplied, followed by the proper IEHMOVE commands for the data set to be loaded (see section C.3).

To link any of the object modules into load form:

```
//DKWICLNK PROC UNIT=2314
//LINK     EXEC PGM=IEWL,PARM=XREF
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSN=DKWIC.INDEXLIB,DISP=(NEW,KEEP),
//         UNIT=&UNIT,SPACE=(TRK,(80,5,2)),VOL=SER=&SER,
```

```
//          DCB=(RECFM=U,BLKSIZE=3400)
//SYSUT1    DD   UNIT=SYSDA,SPACE=(CYL,2)
//SYSLIB    DD   DSN=SYS1.PL1LIB,DISP=SHR
//SYSLIB1   DD   DSN=DKWIC.OBJECT,DISP=OLD,
//          UNIT=&UNIT,VCL=SER=&SER
//          PEND
```

DKWICLNK forms load modules from the object partitions of the data set DKWIC.OBJECT and places them in DKWIC.INDEXLIB. The SER parameter must specify the direct access volume serial number of the previously created data set DKWIC.OBJECT. The load modules will reside on this same volume. The UNIT parameter may be overridden to provide the correct direct access storage class. A LINK.SYSLIN dd statement must be supplied followed by the proper linkage editor control statements to link the desired object modules from DKWIC.OBJECT (see section C.3).

To execute the KWCC DKWIC generator:

```
//KWODKWIC PROC UNIT=2314
//DKWIC    EXEC PGM=KWCDKWIC
//STEPLIB  DD   DSN=DKWIC.INDEXLIB,DISP=SHR,UNIT=&UNIT,
//          VOL=SER=&SER
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//SYSOUT   DD   SYSOUT=A
//SORTIN   DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),
//          DCB=(RECFM=VB,LRECL=&LRECL,BLKSIZE=&BLKSIZE)
//SORTOUT  DD   UNIT=SYSDA,SPACE=(CYL,(2,2)),
//          DCB=*.SORTIN
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,2)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,2)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,2)
//SORTWK04 DD   UNIT=SYSDA,SPACE=(CYL,2)
//          PEND
```

KWODKWIC calls the KWIC DKWIC generator into execution. The SER parameter specifies the volume serial number of the direct access volume containing DKWIC.INDEXLIB. The UNIT parameter may be overridden to provide the correct direct access storage class. A DKWIC.INPUT dd statement must be supplied to indicate the source data to be indexed; a DKWIC.SYSIN dd statement must be supplied to indicate the location of stoplists; a DKWIC.SELECT dd statement locates the actual main term selections; if an authority list is to be used, a DKWIC.AUTHRL dd statement must specify its location. The default parameters for the generation process may be overridden by coding PARM.DKWIC=' parameter list ' (see section C.4). The parameters LRECL and BLKSIZE must be supplied and are described in section C.4.

To execute the KWIC DKWIC generator:

```
//KWIDKWIC PROC UNIT=2314
//DKWIC    EXEC PGM=KWIDKWIC
//STEPLIB  DD   DSN=DKWIC.INDEXLIB,DISP=SHR,UNIT=&UNIT,
//         VOL=SER=&SER
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//SYSOUT   DD   SYSOUT=A
//PRIME    DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//SECNDRY  DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//SORTIN   DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//SORTOUT  DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,2)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,2)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,2)
//SORTWK04 DD   UNIT=SYSDA,SPACE=(CYL,2)
//INDEX    DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//MASTER   DD   SYSOUT=A
//MARKS    DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//         PEND
```

KWIDKWIC calls the KWIC DKWIC generator into execution. The SER parameter specifies the volume serial number of the direct access volume containing DKWIC.INDEXLIB. The UNIT parameter may be overridden to provide the correct direct access storage class. A DKWIC.INPUT dd statement must be supplied to indicate the source data base to be indexed; a DKWIC.SYSIN dd statement locates the stoplists; if an authority list is used, a DKWIC.AUTHRL points to the data set containing the word control list. The default execution time parameters for the index generation process may be overridden by coding PARM.DKWIC=' parameter list ' (see section C.5).

To generate an authority list from a source data set to be indexed:

```
//AUTHRL  PROC UNIT=2314
//DKWIC   EXEC PGM=AUTHLIST
//STEPLIB DD   DSN=DKWIC.INDEXLIB,DISP=SHR,UNIT=&UNIT,
//         VOL=SER=&SER
//SYSPRINT DD  SYSOUT=A
//         PEND
```

AUTHRL calls the authority list generator into execution. The SER parameter specifies the volume serial number of the direct access volume containing DKWIC.INDEXLIB. The UNIT parameter may be overridden to provide the correct direct access storage class. A DKWIC.INPUT dd statement is required to indicate the source data base to be indexed; a DKWIC.SYSIN dd statement locates

the authority list exception tables; a DKWIC.AUTHRL dd statement identifies the location of the authority list to be created; a DKWIC.TITLE dd statement specifies the data set on which the data base, converted to internal form, is placed. The default execution time parameters for the authority list construction can be overridden by coding PARM.DKWIC=' parameter list (see section C.6).

C.3 Installing The DKWIC Indexing Subsystems

The simplest installation of the DKWIC indexing subsystems is to use the load module provided on the distribution tape. To install this system, the following JCL model can be employed:

```
//...      JOB
           the JCL procedures of section C.2
//MOVLIB   EXEC DKWICLD,SER=SYSLIB,UNIT=2314
//LOAD.SYSIN    DD *
COPY PDS=DKWIC.INDEXLIB,TO=2314=SYSLIB,FROM=2400=(DKWIC,12)
/*
//
```

Assumptions:

1) the direct access storage to be used are 2314's (the blocking is such that 2311's may be substituted)

2) the PDS DKWIC.INDEXLIB is placed on the volume named SYSLIB (change name as appropriate) and this volume has at least 80 tracks (in the case of 2314) of available space, and does not already contain a data set named DKWIC.INDEXLIB.

The SER and UNIT parameters and the TO=unit=ser should be changed to those names used by the particular installation.

Once DKWIC.INDEXLIB has been loaded, the indexing procedures of section C.2 can use this data set as a step library.

Should any of the source modules be changed or a new data base interface be written, some of the modules may require recompilation and linkage editing. The first step of this process should be loading the object partioned data set. The following JCL model can be employed:

```
//...       JOB
            the JCL procedures of section C.2
//MOVOBJ    EXEC DKWICLD,SER=SYSLIB,UNIT=2314
//LOAD.SYSIN    DD *
COPY PDS=DKWIC.OBJECT,TO=2314=SYSLIB,FROM=2400=(DKWIC,11)
/*
//
```

Assumptions:

1) the direct access storage to be used are 2314's (the blocking is such that 2311's may be substituted)

2) the PDS DKWIC.OBJECT is placed on the volume named SYSLIB (change name as appropriate) and this volume has at least 32 tracks (in the case of 2314) of available space, and does not already contain a data set named DKWIC.OBJECT.

The SER and UNIT parameters and the TO=unit=ser should be changed to those names used by the particular

installation.

In order to replace one of the members of the DKWIC.OBJECT data set, the member to be replaced must first be scratched and then added to the data set. The following JCL model first scratches the members DKWIC3 and DKWIC4 and recompiles them from the distribution tape:

```
//...       JOB
            the JCL procedures of section C.2
//SCRATCH   EXEC PGM=IEHPROGM
//SYSPRINT  DD   SYSOUT=A
//DD1       DD   UNIT=2314,DISP=OLD,VOL=SER=SYSLIB
//SYSIN     DD   *
SCRATCH DSNAME=DKWIC.OBJECT,VOL=2314=SYSLIB,MEMBER=DKWIC3
SCRATCH DSNAME=DKWIC.OBJECT,VOL=2314=SYSLIB,MEMBER=DKWIC4
/*
//ASM3      EXEC DKWICASM,LABEL=3
//CMP.SYSGO      DD DSN=DKWIC.OBJECT(DKWIC3),
//          DISP=(MOD,KEEP),UNIT=2314,VOL=SER=SYSLIB
//COMP4     EXEC DKWICOMP,LABEL=4
//CMP.SYSLIN     DD DSN=DKWIC.OBJECT(DKWIC4),
//          DISP=(MOD,KEEP),UNIT=2314,VOL=SER=SYSLIB
//
```

Assumptions:

1) the direct access storage used are 2314's (the blocking is such that 2311's may be substituted)

2) the PDS DKWIC.OBJECT exists on the volume named SYSLIB

The relationships between the object and execution forms of the programs is given below to direct the linkage editing required.

DKWIC.INDEXLIB name	DKWIC.OBJECT partition names required	description of load module
KWODKWIC	DKWIC1,DKWIC2,DKWIC3	KWOC DKWIC index generator
AUTHLIST	DKWIC4,DKWIC2,DKWIC3	authority list generator
KWIDKWIC	DKWIC5	KWIC DKWIC index monitor
NEWDKWIC	DKWIC6,DKWIC2,DKWIC3	maxima main term generator
SELECT	DKWIC7	actual main term selection
MASK	DKWIC8	modify maximal main terms
MERGE	DKWIC9	create actual subordinate terms
PRINT	DKWIC10	print DKWIC index

The following JCL model may be used to create part or all of the data set DKWIC.INDEXLIB from object modules:

```
//...    JOB
       the JCL procedures from section C.2
//LINKLIB      EXEC DKWICLNK,UNIT=2314,SER=SYSLIB
//LINK.SYSLIN  DD *
INCLUDE SYSLIB1(DKWIC1,DKWIC2,DKWIC3)
NAME    DKWIC(R)
INCLUDE SYSLIB1(DKWIC4,DKWIC2,DKWIC3)
NAME    AUTHLIST(R)
INCLUDE SYSLIB1(DKWIC5)
NAME    ATODKWIC(R)
INCLUDE SYSLIB1(DKWIC6,DKWIC2,DKWIC3)
NAME    NEWDKWIC(R)
INCLUDE SYSLIB(DKWIC7)
NAME    SELECT(R)
INCLUDE SYSLIB1(DKWIC8)
NAME    MASK(R)
INCLUDE SYSLIB1(DKWIC9)
NAME    MERGE(R)
```

```
INCLUDE SYSLIB1(DKWIC10)
NAME    PRINT(R)
/*
//
```

Assumptions:

1) the direct access storage used are 2314's (the blocking is such that 2311's may be substituted)

2) the data set DKWIC.OBJECT exists on the volume named SYSLIB and all 10 members are present

3) the data set DKWIC.INDEXLIB does not exist on the volume named SYSLIB but will be created by this job.

If only a portion of the load modules are to be created only those particular INCLUDE and NAME statements need to be retained. If DKWIC.INDEXLIB already exists, the SYSLMOD dd statement of the procedure may be overridden by inserting the following dd statement after the EXEC card:

```
//LINK.SYSLMOD  DD DSN=DKWIC.INDEXLIB,DISP=(MOD,KEEP),
//       UNIT=2314,VOL=SER=SYSLIB
```

C.4 The KWOC-DKWIC Hybrid Index Generator - Documentation

The KWOC-DKWIC index generator is divided into three logical segments. The user has the freedom to select or bypass either of the last two.

The initialization phase is always executed where variable length storage requirements are determined and allocated. The stoplists and the authority list, if present, are brought into core and sorted.

If phase 1 is executed, all potential main terms are generated from the source titles after the title words found on the authority list have been replaced by appropriate preferred words. The potential main term file is alphabetically sorted and searched for identical potential main terms. The PMT and its occurrence frequency are printed during this phase in preparation for actual main term selection which occurs in phase 2.

If phase 2 is entered, the sorted potential main term file and the associated statistics file must be available. During this phase, the actual main terms are selected from the PMT file by matching sequence numbers input through a selections file. If no selections file is provided, all PMT are chosen for the final index. As selections are being processed, the PMT statistics file is interrogated to determine when subordinate entries should be permuted. When either all selections have been made or the PMT file is exhausted, the final index is sorted first by the actual main term then by the first words of each subordinate entry. The sorted index records are then passed to a formatting routine where the index is printed according to user specifications.

C.4.1 KWOC-DKWIC Execution Parameters

To allow the index analyst maximum flexibility in generating indexes, several parameters can be supplied during execution to tailor the index generator to his

specific needs. All parameters are found in the PARM field of the EXEC statement (see C.4.5 for exact placement). The format of the parameter string is

```
PARM='phases,delimiters,#terminal,lencode,maxchar,
      maxword,minpmt,maxpmt,lenpage,lenline,
    threshold,autostop,maxstoplen,maxstopwid,
      sortsize,firstpage,#columns'
```

or

```
PARM=D
```

where

phase - two digit number, NM, directing the program to execute the phases indicated;

N=0 - bypass phase 1;

N=1 - create potential main terms using temporary files. At the termination of phase 1, the collated potential main terms reside on the data set named by the ddname SORTOUT. The data set named by the ddname SORTWK01 contains the tally data printed with the potential main term list. These data sets will be destroyed if phase 2 is entered directly;

N=2 - create the potential main terms, copying the files necessary for phase 2 onto permanent data sets. At the termination of phase 1, the potential main terms will reside in the data set named by the ddname SAVEFILE and the tally data concerning like potential main terms resides in the data set named by the ddname TEMPFILE.

N=3 - perform the same function as N=1 except do not print the PMT list;

M=0 - bypass phase 2

M=1 - perform main term selection from temporary files, destroying both potential main term and tally data sets in the process; create and print the final index;

M=2 - perform main term selection from permanent files. Potential main terms are selected from the data set named by the ddname SAVEFILE in conjunction with the tally data set named by the ddname TEMPFILE. Create and print the final index;

M=3 - perform the same function as M=1 except do not print the index but calculate the line estimates only;

M=4 - perform the same function as M=2 except do not print the index but calculate the line estimates only;

Default 10.

Delimiters - varying length character string;
the string of alphanumeric characters which make up both the terminal and non-terminal word delimiters; terminal characters precede non-terminal characters;

default ' '.

#terminal - integer;
the number of characters in the terminal delimiter set;

default 0.

Lencode - integer;
the number of characters in the accession number of the title data being processed;

default 0.

Maxchar - integer;
the maximum number of characters expected in a title phrase;

default 256.

Maxword - integer;
the maximum number of words expected per title;

default 50.

Minpmt - integer;
the fewest number of words in a potential main term;

default 1.

Maxpmt - integer;
the maximum number of words in a potential main term;

default 1.

Lenpage - integer;
the number of lines per page;

default 60.

Lenline - integer;
the number of characters per line; minimum 20 maximum 132;

default 132.

Threshold - integer;
the maximum number of subordinate entries posted beneath a main term in the KWOC-type format;

default 1.

Autostop - integer;
the maximum number of characters in a word that is automatically assumed to belong to the secondary stoplist;

default 2.

Maxstoplen - integer;
the maximum number of locations to be reserved for both the primary and secondary stoplists;

default 0.

Maxstopwid - integer;
the maximum number of characters found in a stoplist word;

default 0.

Sortsize - integer;
the number of 1024 bytes of storage to be used for sort buffer area;

default 20.

Firstpage - integer;
the number of lines to be printed on the first page so that header information can be inserted; omit this parameter if the first page is to be handled in the

same manner as others;

#columns - integer;
the number of columns making up the first page; used in conjunction with firstpage to create a short first page;

The second form of the PARM field permits parameters to be read from the data set PARM. This data set must contain the parameter string of the first form omitting the "PARM=".

The parameters found in the PARM field mentioned above are distinguished only by their position in the parameter string. If the default value of any parameters are accepted, the user must indicate the omission by a comma; the position of omitted parameters is not necessary when the omissions fall to the right of the last parameter present in the list. In the example below,

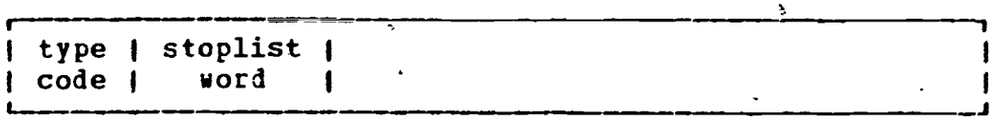
PARM=',,''.,/'',2,6,,,126'

the delimiters consist of ".,/" of which the first two are terminal; the accession code length is 6; the page length is 126; all other parameters assume their default values. Note that all character strings are enclosed in apostrophes; to represent an apostrophe, two consecutive apostrophes must be coded.

C.4.2 Input Of Stoplists To The KWOC DKWIC Index Generator

Both the primary and secondary stoplists are input to the program through the data set associated with the ddname SYSIN. Any word input as a member of the secondary stoplist is assumed also to reside on the primary stoplist. The

records of this file are assumed to be 80 characters in length with one stoplist word per record. The format of a stoplist record is shown is Figure C.1. The type code, a



type code
01 primary stoplist
02 secondary stoplist

FIGURE C.1 STOPLIST ENTRY FORMAT

two digit numeric, indicates the stoplist into which the designated word is placed; code 01 indicates primary; code 02 indicates secondary. Immediately following the type code in the third byte of the record begins the stoplist word itself. The next maxstopwid characters make up the stoplist word. If the word has fewer characters than the maximum, then the word must be padded with blanks. If the word is longer than the maximum specified, only the first maxstopwid characters are used. The number of stoplist records must not exceed the maximum number specified in the PARM statement. If the maximum is exceeded an error message is printed and processing continues ignoring any remaining stoplist words. The stoplist words may appear in any order. They are separated, sorted, and displayed for verification.

C.4.3 Selecting Actual Main Terms For A KWOC-DKWIC Index

Phase 2 of DKWIC index generation requires the index analyst to choose those main terms that are to appear in the final index. From the output of phase 1, a list of sequence numbers corresponding to the chosen main terms is prepared. These sequence numbers are punched into cards in free format (i.e. at least one blank between numbers) in ascending order and presented for input in the data set identified by the SELECT ddname. If this dd statement is omitted, all potential main terms are selected.

C.4.4 Job Control For A KWOC-DKWIC Index Generation

Below is a list of all ddnames and the required attributes of the data sets used by the program. Note that several data sets may be optionally supplied.

ddname	usage
SYSPRINT	sequential output data set on which all messages and the final index are placed.
INPUT	sequential input data set on which resides the title data to be indexed.
AUTHRL	sequential input data set on which the authority list resides. This statement is present only when the authority list is used.
SYSIN	sequential input data set on which reside the communication record with the interfacing subroutine (see section C.7) and stoplists.
PARM	optional sequential input data set which contains the parameters for the index generation when the PARM=D is specified.

SELECT sequential input data set used during phase 2 to input the sequence number denoting the actual main terms. If this dd statement is omitted, then all potential main terms are selected if phase 2 is entered.

SAVEFILE sequential data set on which the potential main terms are copied during phase 1 only when the first digit of the phase parameter is 2. This dd statement defines the input potential main term data set when the phase 2 option is set to 2 or 4. (LRECL=MAXCHAR+LENLINE/2+LENCODE+55, BLKSIZE=N*LRECL+4)

TEMPFILE sequential data set on which the tally of like potential main terms are placed by phase 1 when the first digit of the phase parameter is set to 2. During phase 2 this data is used to input the tally information if the second digit of the phase parameter is set to 2 or 4.

SORTLIB the system sort library. DSN=SYS1.SORTLIB,DISP=SHR

SORTIN sequential data set which is used as a temporary input/output file during sorting; LRECL and BLKSIZE must be identical to SAVEFILE.

SORTOUT temporary input/output data set used for sorting procedures. LRECL and BLKSIZE should be identical to SAVEFILE.

SYSOUT sequential output message data set required for the SORT/MERGE program.

SORTWKOn work areas for the sort routine. (n=1,2,3 minimum)

C.4.5 Sample JCL For A KWOC-DKWIC Index Generation

```
//...      JOB
the JCL procedures from section C.2
//GEN       EXEC KWODKWIC,UNIT=2314,SER=SYSLIB,
//          DKWIC.PARM='parameter list',
//          LRECL=described above,
//          BLKSIZE=n*LRECL+4
//DKWIC.INPUT  DD *
            title data to be indexed
//DKWIC.AUTHRL DD DSN=AUTHRL,DISP=OLD,
//          UNIT=2314,VOL=SER=SYSLIB
//DKWIC.SYSIN  DD *
            interface control card
            stoplists
```

```
//DKWIC.SELECT DD *
        sequence numbers of selected entries
//
```

C.4.6 Messages Issued By The KWOC-DKWIC Index Subsystem

```
DKWIC.00 - VERSION cc - d
   PHASES                  dd
   DELIMITERS
   GROUP1                  cccc
   GROUP2                  cccc
   ACCESSION LENGTH        dd
   MAXIMUM TITLE (CHAR)    ddd
   MAXIMUM WORDS           ddd
   MIN PMT                 d MAX PMT d
   PAGE LENGTH             ddd
   PAGE WIDTH              ddd
   PERMUTATION THRESHOLD   d
   AUTOMATIC STOP          d
   STOPLIST
   WIDTH                   dd
   MAXLEN                  dd
```

the parsing of the parameter field is displayed for verification.

DKWIC.01 - LINE WIDTH ERROR

the lenline parameter was greater than 132 or less than 20; the line width is set to 132 and processing continues.

DKWIC.02 - NUMBER GROUP1 CHARACTERS > SIZE OF DELIMITERS

the number of characters found in the delimiter string was less than #terminals; all characters in the delimiter string are assumed to be terminal; processing continues.

DKWIC.03 - MIN NUMBER WORDS/MAIN TERM > MAX

the minimum number of words specified to be in a potential main term is greater than the maximum specified; the minimum number is set to the maximum and processing continues.

DKWIC.04 - STOPLIST GREATER THAN LENGTH SPECIFIED

the number of stoplist words found in the SYSIN data set was greater than the number expected. Only the first maxstoplen are considered.

DKWIC.05 - PROGRAM ERROR, ONCODE=DDDD

a terminal execution error has been found by the PL/I error handler. The ONCODE is listed and a PL1DUMP is initiated if a PL1DUMP dd card is present.

DKWIC.06 - TOO MANY CHARACTERS IN RECORD - dddd

the number of characters in the title whose accession code is dddd is greater than maxchar. The title is ignored and processing continues.

DKWIC.07 - TOO MANY WORDS IN TITLE TO PROCESS - dddd

the number of words in the title whose accession code is dddd is greater than maxword. The title is ignored and processing continues.

DKWIC.08 - SORT ERROR

the SORT/MERGE program returned a condition code other than zero. The sort control cards are listed below this message. Consult the message data set SYSOUT for details concerning the error. Execution terminates.

DKWIC.10 - PHASE 1 RESULTS

TITLES	dddd
WORDS	dddd
WORDS/TITLE	dddd
1-STOPLIST	dddd
2-STOPLIST	dddd
TOTAL PMT	dddd
UNIQUE PMT	dddd
TOTAL PMT/TITLE	dddd
CHARACTERS/TITLE	dddd
CHARACTERS/REM TITLE	dddd

phase 1 has been completed and the results are posted for verification.

DKWIC.20 - PHASE 2 RESULTS

ACTUAL MAIN TERMS	dddd	dd.dd
PERMUTED TYPE		
#TITLES	dddd	dd.dd
#ENTRIES	dddd	dd.dd
#LINES	dddd	dd.dd
KWOC-TYPE		
#TITLES	dddd	dd.dd
#ENTRIES	dddd	dd.dd
#LINES	dddd	dd.dd

phase 2 has been completed and the results are displayed for inspection. The statistics are grouped by the type of entry; each entry is given as the raw number of occurrences and the percentage of occurrences in the final index.

```
DKWIC.30 - SIZE ESTIMATES - LINEWIDTH ddd - PAGE ddd
TITLES/ENTRY  MAIN TERMS  EST KWOC  EST DKWIC
   d            dd          ddd       ddd
   d            dd          ddd       ddd
   .            .           .         .
   .            .           .         .
```

The number of main terms (MAIN TERMS) having N titles (TITLES/ENTRY) is displayed along with an estimate of the number of lines in the index these entries will produce if the entry is formatted as a KWOC-type (EST KWOC) or DKWIC-type (EST DKWIC). The linewidth and pagesize are also printed for reference when making calculations of the number of pages of index.

C.4.7 KWOC-DKWIC Index Subsystem Implementation Restrictions

The KWOC DKWIC generator operates under full OS/360 operating system. The program is written in PL/I version 5.2 and requires a minimum of 126K bytes of core to operate effectively. If the stoplists and authority list become exceedingly large, this minimum will not be sufficient. The program direcctly calls the system 360 SORT/MERGE facility to handle variable length record sorts.

C.5 The KWIC-DKWIC Hybrid Index Generator - Documentation

The KWIC DKWIC generator produces an index through the execution of five phases, implemented as PL/I subprograms called by an assembly language submonitor. Each of these phases may be selected or bypassed under user control.

In the first step, all maximal main terms are generated from the data base. The specificity of each MMT as well as

each specificity unit boundary is written with each record. These records are tagged with an internal sequence number which represents the relative record position of the title which is kept in internal format in another file. A data set of pointer records is also generated for this title file which contains information to locate all words in the corresponding title and indications of stoplist characteristics. The maximal main term file is then sorted alphabetically and passed to the selection program.

The maximal main term file is passed sequentially by the selection program where MMT statistics are gathered and the PMT tree is built for each MMT beginning with the same initial word. After each tree is built, it is examined for maximum and minimum posting criteria. At this time pointers into the MMT file are created accompanying the actual specificity and count of the number of titles containing the actual main term.

The transformation of the maximal main terms to actual main terms occurs in the next step where the MMT file, the specificity and occurrence files are passed in parallel. Each maximal main term is reduced to the specificity indicated by the corresponding pointers. The user supplied subordinate permutation threshold is matched with the frequency of occurrence of each main term and a marker concerning this decision is placed in the actual main term record before it is written on a main term file. The main

term file is then sorted by the internal title sequence number.

The title and associated pointer files are read in parallel matching internal sequence numbers against those present in the main term file. A match signifies the need to form a subordinate entry from the corresponding title. When the number of occurrences of this main term phrase falls below the permutation threshold, the title is rotated so the initial word of the main term entry appears as the first word of a KWIC-type entry. When the threshold is exceeded, all occurrences of the main term are extracted from the title. Subordinate entries are generated beginning with each word that remains in the title and is not a member of the secondary stoplist. When all AMTs have been processed, control passes to a program which sorts the main and subordinate entries.

The sorted entry file is then formatted by a print routine which examines first the permutation marker to indicate whether a KWIC or DKWIC subordinate entry should be used. The index entry is then printed according to user specifications.

C.5.1 KWIC-DKWIC Execution Parameters

The execution of each of the phases of the KWIC DKWIC generator is governed by an execution monitor written in IBM/360 assembly language. This monitor accepts several keyword parameters which supply the necessary variable

information for tailoring the programs to generate a specific index. These parameters appear on the PARM field of the EXEC statement invoking the DKWIC indexing program and take the following form:

BRKLIST - varying length character string
The set of break characters to be used to discern word boundaries in the titles being indexed. The first character is used to delimit the remainder of the break characters and can be any character not found in the list. The set of terminal break characters must appear first in the list followed by the non-terminal ones. The break character delimiter separates these strings as well as ends the non-terminal list. Thus, if ",.:;" are terminal and "/-" are non-terminal, then the breaklist is written as

Q,.:;Q/-Q

where the breaklist delimiter is Q. The breaklist is a positional parameter and must appear first in the PARM field. If the entire list is omitted, it must be represented by a comma. Two successive breaklist delimiters are interpreted as a null string. Default QQ Q denoting no terminal break characters with a blank being the only non-terminal. A blank is automatically supplied to the user even when a breaklist is specified.

Default QQ Q

CODE=lencode
the length of the accession code;

default CODE=0

SPEC=maxspec
the maximum specificity of a maximal main term;

default SPEC=3

STOP=(autostop,stopwidth,maxstoplen)

Autostop - the maximum number of characters automatically assumed to be members of the secondary stoplist

Stopwidth - the number of characters in the longest stoplist word

Maxstoplen - the maximum number of words expected on the stoplist

Default STOP=(2,0,0)

POST=(maxpost,minpost)

Maxpost - the maximum number of titles to be posted at a particular specificity

Minpost - the minimum number of titles to be posted at a particular specificity

Default POST=(4,2)

PAGE=(linewidth,pagelength,reserved,numcol)

Linewidth - the number of characters per line

Pagelength - the number of lines per page

Reserved - the number of lines (full page width) reserved on the first page of the index. This parameter allows the user to print a short first page.

Numcol - the number of columns expected on the first page

Default PAGE=(132,60,0,0)

PERM=threshold

threshold - the maximum number of titles forming a group of similar main term entries which will be posted as KWIC entries in the final index.

Default PERM=2

FORM=(pages,chars/col,colsep,res,orig,min,max,wid,len)

The FORM parameter is used to specify automatic formatting specifications. If this parameter is present, the PERM and PAGE parameters need not be specified since those parameters are calculated by the automatic formatting routine.

Pages - the maximum acceptable number of pages allowed for the index. The numeric specified must include partial first and last pages.

Chars/col - the minimum acceptable number of characters per line per column in a printed entry in the final index. This numeric includes the number of characters in the accession code but does not include the number of blank characters between columns.

Colsep - the number of blank characters to be inserted between columns when the final index is prepared for photoreduction.

Res - the number of lines (full page width) to be reserved on the first page of the index. This parameter allows the user to print a short first page.

Orig - an integer between 0 and 100 which represents the minimum acceptable percent of original size for the final index.

Min - the minimum acceptable permutation threshold.

Max - the maximum acceptable permutation threshold.

Wid - the width of the field in 10ths of an inch onto which the photoreduced copy of the index is to be fitted.

Len - the length of the field in 10ths of an inch onto which the photoreduced copy of the index is to be fitted.

Default FORM=(0,50,5,0,60,2,20,75,100)

PHASE=execphase

an integer representating the phases to execute
1 - phase 1
2 - phase 2
4 - phase 3
8 - phase 4
16 - phase 5
execphase is the sum of all or any of these quantities. The phases are always executed in order.

Default PHASE=31

With the exception of BRKLIST, the parameters are keyword oriented and can appear in any order. The multiple arguments of keyword parameters are positional. If the default values of these parameters are to be assumed, their position must be indicated by a comma. For example, to change just pagelength, the PAGE parameter is coded

PAGE=(,120)

The first two letters of any keyword can be used as abbreviations of any of the parameters mentioned above.

If the parameter field is too large to fit onto the EXEC card, substitute the word CARD for the parameter list. The parameter field is then read from up to the first two card images of the data set associated with the ddname PARM. The parameters are punched in the same keyword format described above, dropping the opening and closing apostrophes.

C.5.2 Input Of Stoplists To The KWIC-DKWIC Index Generator

The stoplists for the KWIC DKWIC generator are input in the same manner and form as the KWOC DKWIC process (see C.4.2).

C.5.3 Job Control For KWIC-DKWIC Index Generation

Below is a list of all ddnames and the required attributes of the data sets used by the program. Note that several data sets may be optionally supplied.

DDNAME	USAGE
SYSPRINT	sequential output message data set
SYSIN	sequential input data set from which the data-base interface control and stoplists are read (LRECL=80)
INPUT	sequential input data set from which the data base of titles is read
AUTHRL	optional sequential input data set on which resides the authority list created by the word transformation routine
PRIME	sequential data set on which the titles in internal format are placed for later reference (LRECL=304,BLKSIZE=3348,RECFM=VB)
SECNDRY	sequential data set on which pointers to all words found in the corresponding PRIME title record is placed for later use (LRECL=144,BLKSIZE=1440, RECFM=FB).
SORTIN	sequential data set which is used as input to the standard sort package. This data set is used by three of the phases for output, changing the RECFM, LRECL, and BLKSIZE characteristics each time. Do not specify DCB characteristics for this file.
SORTOUT	sequential data set which is used to hold the output from the sort program. This data set is used as input to four phases of the indexing operation and should not contain DCB characteristics.
SORTWKOn	sequential data sets defining sort work areas (n=1,2,3 minimum). The statistics for the PMT tree are kept on one of these data sets.
SORTLIB	the sort library for the standard sort program. The index generator requires exits E15 and E35.
SYSOUT	sequential output message data set used by the sort routine
MARK	temporary data set used to hold the selection markers generated by the auto-select routine.
INDEX	sequential output data set onto which the final

index is placed prior to formatting.

MASTER sequential output data set onto which the final formatted index is placed.

PARM optional input data set describing an alternate parameter list input stream

C.5.4 Sample JCL For KWIC-DKWIC Index Generation

```
//...        JOB
           the JCL procedures of section C.2
//ADKWIC     EXEC    KWIDKWIC,
//          PARM.DKWIC=' parameter list '
//DKWIC.SYSIN      DD *
           interface control card
           stoplists
//DKWIC.INPUT      DD *
           data base cards
//
```

C.5.5 Messages Issued By The KWIC-DKWIC Index Subsystem

```
DKWIC.00 - DKWIC INDEX - VERSION V - n
BREAK CHARACTERS
TYPE 1            1111
TYPE 2            1111
CODE LENGTH       nnn
MAX SPECIFICITY   nnn
AUTOMATIC STOP    nnn
WIDTH             nnn
MAXLEN            nnn
```

An echo of the parameters input to phase 1 are presented for verification.

DKWIC.01 - STOPLIST GREATER THAN LENGTH SPECIFIED

The maxlen parameter specified a number less than the total number of words presented for the entire stoplist. Execution continues with the first maxlen stoplist words.

DKWIC.02 - TITLE RECORD IGNORED; LENGTH EXCEEDS MAX

A title record containing more than 300 characters including accession code has been found an is printed under this message. The title record is ignored and processing continues.

DKWIC.03 - TITLE RECORD IGNORED; MAX WORDS/TITLE EXCEEDED

A title record containing more than 32 words has been detected and printed below this message. The title has been ignored and processing continues.

DKWIC.04 - PROGRAM ERROR, ONCODE = nnnn

A serious error has occurred during the execution of the program. The condition is described by the oncode numeric. This message usually follows a more descriptive error indication printed by the PL/I error handler. In event the error handler abnormally terminates, the error can be determined by consulting the PLI Reference Manual for oncode conditions.

```
DKWIC.05 - MMT STATISTICS
NUMBER OF TITLES        nnnn
NUMBER OF WORDS         nnnn
WORDS ON SEC STOP       nnnn
WORDS ON PRIM STOP      nnnn
1-ARY MAX MAIN TERMS    nnnn
2-ARY MAX MAIN TERMS    nnnn
    .                     .
    .                     .
    .                     .
```

The statistics for MMT generation are presented for the user. This message is printed during the final step of phase 1.

```
DKWIC.10 - SELECTION CRITERIA
MAX POSTING             nn
MIN POSTING             nn
```

The maximum and minimum posting limits are displayed for user verification as phase 2 is entered.

```
DKWIC.12 - SELECTION STATISTICS
PMT TREES               nnnn
1-ARY MT                nnnn
2-ARY MT                nnnn
  .                       .
  .                       .
  .                       .
```

The statistics for the selection phase are presented for the user. The number of PMT trees examined and the number of selections made at each MT specificity is displayed.

DKWIC.13 - INDEX SIZE ESTIMATES

TITLE/GROUP OR THRESHOLD	NUMBER GROUPS	EST KWIC LINES	EST DKWIC LINES	TOTAL EST LINES
1	nnn	nnn	nnn	nn.n
2	nnn	nnn	nnn	nn.n
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

An estimate of the size of the index to be printed is displayed. The number of main terms contained in precisely n titles is found in the nth entry under TITLE/GROUP if the threshold is n then the number of titles which will form DKWIC-type entries and KWIC-type entries are displayed beneath EST DKWIC LINES and EST KWIC LINES respectively. From the averages concerning AMT specificity, words/title, and secondary stoplist criteria, an estimate of the number of lines in the index is presented for each threshold value.

DKWIC.40 - DKWIC ENTRY LARGER THAN MAX

An entry has been generated which exceeds the maximum record length. The record, displayed below this message, is ignored and processing continues. The number of characters in this record after the main term has been extracted must be shortened to be accepted.

C.5.6 KWIC-DKWIC Index Subsystem Implementation Restrictions

1) A maximum of 300 characters has been allocated for any title of the data base and any index item temporarily stored by the program. The program detects this condition and ignores such records informing the user of the action.

2) A single title cannot contain more than 32 words as defined by the word delimiter set. The program detects this condition and ignores such records informing the user of the action.

3) All maximal main terms are truncated to 50 characters without warning.

4) The program requires 126K bytes of core to execute effectively. When large stoplists and authority lists are used, 126K bytes may be inadequate.

5. The programs operate under full OS/360 and directly call the system sort package for fixed and variable length record sorts.

C.6 The Authority List Generator - Documentation

The word transformation routine is embodied in a program separate from any indexing routines and is intended to be executed as a preprocessor of the titles being indexed. The inputs consist of the data base and appropriate exceptions lists; the output, the authority list ready to be used by the indexing routines.

C.6.1 Authority List Execution Parameters

To effect generality, several parameters regarding the estimate of array and string sizes are made available to the user so as not to limit the usefullness of the program. The parameter list must be supplied on the EXEC card describing the authority list generator. It is of the form:

PARM='LCODE,BRKLIST,LISTLEN'

where

LCODE - integer
length of the accession code of this data base

default 0

BRKLIST - character string
a list of the characters to be used as word delimiters

default ' '

LISTLEN - integer
the maximum number of words expected on the authority list.

default 100

The parameters found in the PARM field are distinguished by their position only in the parameter string. If the default value of any parameters are accepted, the user must indicate the omission by a comma; the positions of omitted parameters is not necessary when the omissions fall to the right of the last parameter present in the list. Character strings included in the PARM field must be enclosed by pairs of apostrophes.

C.6.2 Authority List Exceptions List Input

All exception lists are entered through the SYSIN data set. Each exception list word is punched, one word per card, following a two byte numeric list code (see Figure C.3 for code numbers and designations). The words must be grouped by exception list code; the words within a single exception list can be placed in any order (see Figure C.2).

The first record of the exception list holds two positional parameters which direct storage allocation for the lists. These parameters are:

MAXEXCEPT,WIDEXCEPT

where

MAXEXCEPT - integer > 0

maximum number of words expected for all exception list words

'D' .CEPT - integer > 0
maximum number of characters expected in the longest exception list word

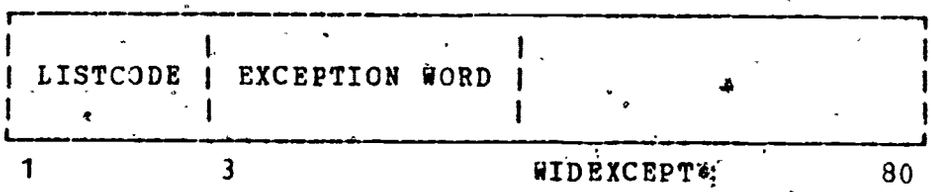


Figure C.2 Exception list format

A review of the exception list definitions and their assigned code numbers are displayed in Figure C.3.

01 - non-transformable words ending in "consonant-s" (e.g. physics, MEDLARS, etc.).

02 - non-transformable words ending in "vowel-s" (e.g. atlas, pathos, etc.) excluding those ending in "sis".

03 - non-transformable words ending in "ies" (e.g. series, etc.).

04 - irregular plurals ending in "es" whose singulars are not formed by dropping the final "s" (e.g. indices, etc.).

05 - corresponding singular entry for irregular plurals found on list 04 (e.g. index, etc.).

06 - transformable words ending in "sses" whose singulars are formed by dropping the final "ses" (e.g. busses, etc.).

07 - transformable words ending in "ses" whose singulars are formed by dropping the final "es" (e.g. thesauruses, chorsuses, etc.).

Figure C.3 A synopsis of the exception list codes and their definitions

C.6.3 Authority List Format

The authority list produced by this program is an array of the singular and plural words transformed by the word transformation routine. Each element of the array is in one of two formats, regular preferred word and irregular preferred word.

The 18 bytes of a regular preferred word entry contains the singular or plural word which is used to match words in the data base (see Figure C.4). When a match is found, the preferred word is formed by concatenating the preferred word

stem (whose offset is given as a binary integer in the last two bytes of the authority list entry) with an ending chosen from an array whose subscript is stored in the "ending indicator" byte (see Figure C.4).

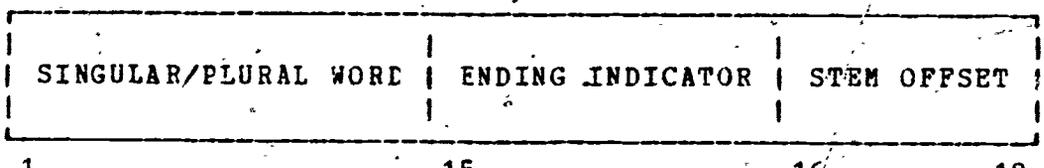


Figure C.4 Regular preferred word format

The "ending indicator" is a one byte binary integer pointing into an "ending" array (see Figure C.5). The preferred word for each entry of the authority list is formed by concatenating the word stem with the appropriate ending. If the word ACTIVITIES appeared in the data base, both the words ACTIVITY and ACTIVITIES would appear in the authority list. The "stem offset" of each entry would be 7 and the "ending indicator" would be 4. The preferred word generated would be ACTIVITY(IES) for both the singular and

ending indicator	ending
1	(S)
2	(ES)
3	(SES)
4	Y(IES)
5	IS(ES)
6	F(ES)

Figure C.5 Endings used to form preferred words

plural concept.

If the preferred word stem cannot be generated from the singualr or plural word, the "ending indicator" byte contains an asterisk and the "stem offset" bytes are interpreted as a subscript into the authority list pointing to the preferred word. This irregular preferred word format differs from the normal format in that a preferred word code corresponding to the re-interpretation of the "stem offset" precedes the replacement word. The preferred word code is so chosen so that upon sorting of the authority list words, this record will be placed in a position corresponding to this code. An irregularly formed preferred word is handled in the same manner as a regular preferred word once the word stem has been retrieved.

To indicate storage requirements to any program using the authority list, the first record of the list contains in free format the number of words in the list as well as the number of characters in each record.

C.6.4 Job Control For The Authority List Generator

Below is a list of all ddnames and the required attributes of the data sets used by the program. Note that one data set may be optionally supplied.

DDNAME	USAGE
SYSPRINT	sequential output message data set
SYSIN	sequential input data set holding the data-base-interface control card image, the exception list control image, and the exception lists (LRECL=80)
INPUT	sequential input data set holding the titles from which the authority list is built
AUTHRL	sequential output data set upon which the authority list is placed (LRECL=18,BLKSIZE=360)
TITLE	optional output data set on which the titles in internal format are placed

C.6.5 Sample JCL For The Authority List Generator

```
//...        JOB
             the JCL procedures from section C.2
//ALIST      EXEC AUTHRL,
//           UNIT=SYSLIB,
           PARM.DKWIC='parameter list'
//DKWIC.AUTHRL  DD DSN=&&AUTHRL,DISP=(NEW,PASS)
//           SPACE=(360,(10,10)),
//           DCB=(RECFM=FB,LRECL=18,BLKSIZE=360)
//DKWIC.INPUT   DD *
             title data to be indexed
//DKWIC.SYSIN   DD *
             interface control card
             exception list control card
             exception lists
//
```

C.6.6 Messages Issued By The Authority List Generator

DEPLRL.01 - NOT ENOUGH SPACE FOR EXCEPTION LISTS
not enough space was estimated on the exception list control card for the exception lists input. The exception list entries which occur after overflow are ignored. Processing continues.

DEPLRL.02 - NOT ENOUGH SPACE FOR AUTHORITY LIST
not enough space has been estimated for the authority list in the PARM statement. All singular entries marked with an asterisk (*) have not been added to the list.

DEPLR.03 - THE AUTHORITY LIST REQUIRES ddd LOCATIONS

C.6.7 Authority List Subsystem Implementation Restrictions

1) The maximum number of words found in a title cannot exceed 30. Unpredictable results may occur but processing continues.

2) the maximum number of characters in a title is fixed at 512. Unpredictable results can occur if this limit is exceeded. Processing continues.

3) Authority list entries are restricted to 18 bytes. The singular or plural word is truncated to 15 bytes without warning.

C.7 Interfacing The Data Base

Each indexing subsystem requires that title data be presented to it in a format that is easily manipulated by the index generator. The task of converting external data formats to the internal form used by the generator is assumed by an externally compiled subroutine. Whenever data in a new format requires indexing, only a new interface subroutine is required.

Figure C.6 depicts the format into which all title data must be converted. The first LENCODE bytes of the varying length string contains the accession code for the title which follows immediately. No padding of the title string is necessary. The maximum length of a record is defined for each indexing subsystem.

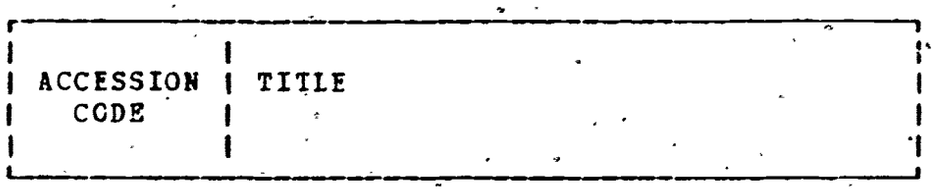


Figure C.6 Internal title format

C.7.1 Requirements Of An Interface Subroutine

To construct an interfacing subroutine, the following conventions must be followed:

1. The subroutine operates as a PL/I function with the following calling sequence and attributes :

```
GETRECORD: PROCEDURE(BUFFER,LENCODE,POINTER)
    RETURNS (BIT(1));
    DECLARE
        BUFFER CHAR(*) VAR,
        LENCODE FIXED BINARY (31),
        POINTER POINTER;
```

BUFFER - character string to be returned containing the accession code and title in internal format.

LENCODE - fixed binary fullword informing the subroutine of the number of characters in the accession code.

POINTER - a pointer variable which upon return contains the address of the next record to be input by the interfacing subroutine.

2. The subroutine must use the ddname INPUT to acquire the title data to be converted. The attributes of INPUT are RECORD INPUT.

3. The first call to the subroutine is for initialization purposes. Therefore, the subroutine must have at least one variable in STATIC storage to indicate the called state. During this call the subroutine may access the first 80 bytes of the STREAM file SYSIN for any variant information concerning the title format.

4. The subroutine returns a yes (RETURN ('1'B)) when BUFFER has been filled with a title. It returns no (RETURN ('0'B)) when no more records are available for processing.

C.7.2 Chemical Titles Interface Subroutine

An interfacing subroutine which converts the Chemical Titles data format to internal form is included with the indexing subsystems. This format was adopted by CHEMICAL ABSTRACTS SERVICE and used for all pre-1971 Chemical Titles source tapes. This subroutine handles titles coded in either the pre-1971 standard file format or the results from a Chemical Titles search.

The standard record contains 80 bytes (Figure C.7) of which the first 17 bytes form the accession code. Column 18 is a type code which indicates how the remainder of the information on the card image is to be intrepted. Within each "type", the records are sequenced in column 19, the "seq" field, beginning with sequence number 1. Type = 1 refers to author records, three authors per card. Type = 2 refers to title records. The title begins in column 21 in the first card. If a second card is necessary, the title

must be broken on a word boundary and continued in column 23 of the next title card. Figure C.8 exemplifies a title in this format.

AUTHOR RECORD

ACCESSION CODE	TYPE 1	SEQ	FIRST AUTHOR	SECOND AUTHOR	THIRD AUTHOR
1	18	19	21	41	61 80

TITLE RECORD

ACCESSION CODE	TYPE 2	SEQ 1	BEGINNING OF TITLE
1	18	19	21 80

TITLE CONTINUATION RECORD

ACCESSION CODE	TYPE	SEQ	IGNORED	CONTINUATION OF TITLE
1	18	19	21	23 80

TYPE
1 - AUTHOR RECORD
2 - TITLE RECORD

Figure C.7 Chemical Titles input format

```
1          17  21          41          61          80
CODENO011    11 AUTHOR1    AUTHOR2    AUTHOR3
CODENO011    12 AUTECR4
CODENO011    21 BEGINNING OF TITLE, NOTE THAT WHEN
CODENO011    22   CONTINUED, THE TITLE IS BROKEN AT
CODENO011    23   A WORD BOUNDARY.
```

Figure C.8 Example of a citation in Chemical Titles format

The Chemical Titles answer format is very similar to the one just described with the exception of the addition of a five byte question number preceeding the standard form and a five byte question weight following.

The interfacing subroutine is capable of merging any record types into a record suitable for indexing. To indicate to the subroutine which types to merge, a nonblank character in the corresponding column of the first record in the SYSIN data set indicates that that type is to be merged. For instance, a character punched into columns 1 and 2 of the first SYSIN record causes the subroutine to concatenate the author and title record types. The first four columns are recognized, type two, three, and four are handled identically. A nonblank character in column five of this same record indicates that the Chemical Title answer format is being used.

The interfacing subroutine replaces trailing blanks of an input record with a single blank before the concatenation of more records. Any blanks found in an author record are replaced by X'FF'. In this manner, the entire author's name

and initials are treated as a single word by the indexing routine. The scan of an author's name is terminated by the occurrence of a contiguous pair of blanks.

C.8 Word Finder Subroutine

An assembly language routine has been implemented to speed the process of finding words in phrases of arbitrary lengths. The routine contains four entry points, three of which are called by the PL/I main program to initialize internal tables before successive calls to the fourth entry yield the information for processing the string, word by word.

The first entry, INITIAL, clears a 256 byte translate table (TABLE) and must be called first by any program using the routine.

```
required declarations
TABLE CHAR(256)
calling sequence
CALL INITIAL(TABLE);
```

The second entry loads the translate table cleared by INITIAL with the word delimiters to be used. The user supplies the delimiters (DELIMITERS) in a varying length character string variable. A one byte character string variable (TYPE) identifies the type of delimiter string input. This character is inserted in the translate table

offset by the hexadecimal equivalent of each character in the delimiter string.

required declarations

```
DELIMITERS CHAR(N) VAR,
TYPE CHAR(1)
```

calling sequence

```
CALL SET(TABLE,DELIMITERS,TYPE);
```

The third entry point is a means of saving some execution time by bypassing some unnecessary dynamic loading of parameter lis s. This entry point is used to pass the parameters concerning the word string to translate and the arrays which contain the pointers to the words in this string so that the fourth entry which performs the word finding operation can be called without parameters.

required declarations

```
BUFFER CHAR(MAXCHARS),
(BREAKTYPE,SECSTOP,PRISTOP) CHAR(MAXWORDS) VAR,
(OFFSET,LENGTHWORD) (MAXWORDS) FIXED BINARY(31),
STOPLIST(MAXSTOP) CHAR(WIDSTOP),
(LSEC,LSTOP,AUTOSTOP) FIXED BINARY(31)
```

calling sequence

```
CALL SETVAR(BUFFER,TABLE,OFFSET,LENGTHWORD,BREAKTYPE,
SECSTOP,PRISTOP,STOPLIST,LSEC,LSTOP,AUTOSTOP);
```

Where

BUFFER - location of the word string to translate

OFFSET - OFFSET(I) contains the location of the first

character of word I in the BUFFER string after translation.

LENGTHWORD - LENGTHWORD(I) contains the length of word I in the BUFFER string after translation.

BREAKTYPE - SUBSTR(BREAKTYPE,I,1) contains the largest delimiter type terminating word I in the BUFFER string after translation.

SECSTOP - SUBSTR(SECSTOP,I,1) contains a one (X'F1') if word I was found on the secondary stoplist; zero (X'F0') otherwise.

PRISTOP - SUBSTR(PRISTOP,I,1) contains a one (X'F1') if word I was found on the primary stoplist; zero (X'F0') otherwise.

STOPLIST - the location of the sorted stoplist. The secondary stoplist must be loaded first into the array followed by the primary stoplist.

LSEC - the actual number of words in the secondary stoplist (the first LSEC words of STOPLIST are assumed to hold the secondary stoplist).

LSTOP - the actual number of words in the stoplist.

AUTOSTOP - the upper limit of the number of characters to be found in a word which is automatically assumed to be on the secondary stoplist.

The word string to be translated must be moved to the location BUFFER before translation can begin. The string is unaffected by any translation process. The lengths of the

varying strings BREAKTYPE, SECSTOP, and PRISTOP reflect the number of words found in the string BUFFER after translation. To retrieve word I, the SUBSTR function is used by the calling program:

SUBSTR(BUFFER,OFFSET(I),LENGTHWORD(I))

This retrieves just the word with no terminating delimiters attached.

The translation algorithm is equipped with a speedy binary search which performs lookups in the array STOPLIST. If the number of characters of a word does not exceed AUTOSTOP, the corresponding locations of SECSTOP and PRISTOP are both set to one. No lookups are performed if the number of characters found in a word exceeds WIDSTOP, the number of characters in each stoplist word. A word found on the secondary stoplist causes the corresponding locations of SECSTOP and PRISTOP to be set to one. Only after a failure of the secondary stoplist search is the primary stoplist searched. If no stoplist lookups are desired, substitute any array for STOPLIST and a fullword binary zero for LSEC and LSTOP. When LSEC is equal to LSTOP, only the first LSEC STOPLIST words are searched.

To initiate the translation of BUFFER, the fourth entry point is used.

Calling sequence

CALL FIND

BIBLIOGRAPHY

Adams,68
Adams,W. And Lockley, L., "Scientists Meet the KWIC Index", American Documentation, 19(1), 47(1968)

Armitage,67
Armitage,J. And Lynch, M., "Articulation in the Generation of Subject Indexes by Computer", Journal of Chemical Documentation 7, 170(1967)

Artandi,68
Artandi, S., An Introduction to Computers in Information Science, Scarecrow Press Inc., Metuchen, N.J., 1968

ASEE,71
Mathis, B., Lasher, R., and Petrarca, A., editors, Participant Index and Subject Index for ASEE Program, 79th Annual ASEE Meeting June, 1971, Annapolis, Md.

Belzer,71
Belzer,J.,"Justification for Automatic Indexing by Frequency Distribution of Words", Journal of the American Society for Information Science, 22(3), 226(1971)

Bottle,70
Bottle, R., "Title Indexes as Alerting Services in Chemical and Life Sciences", Journal of the American Society for Information Science, 21(1), 16(1970)

Brodie,70
Brodie,N., "Evaluation of a KWIC Index for Library Literature", Journal of the American Society for Information Science, 21(1), 22(1970).

Brown,63
Brown,A., editor, Normal and Reverse English Word List, University of Pennsylvania, Philadelphia, 1963

Bush,45
Bush,V., "As We May Think", Atlantic Monthly, 176, 101(1945)

Carroll,69
Carroll, J. And Roeloffs,R., "Computer Selection of Keywords Using Word-Frequency Analysis", American Documentation 20(3), 227(1969)

CAS,72
Chemical Titles, Chemical Abstracts Services, Columbus, Ohio

CCM,72
PANDEX Current Index to Scientific and Technical Literature, CCM Corporation, a subsidiary of Crowell, Collier and MacMillan, Inc., New York, N.Y.

Cheydleur,67
Cheydleur, B., "Indexing Depth, Retrieval Effectiveness and Time Sharing", National Conference on Electronic Information Handling, edited by A. Kent, Thompson Book Co., Academic Press, London, 1967, p37

Citron,59
Citron,J., Hart, L., and Ohlman, H., "A Permutation Index to the 'Preprints of the International Conference on Scientific Information'", Report SP-44, Systems Development Corporation, Santa Monica, California, 1959

Dattola,69
Dattola, R., "Fast Algorithm For Automatic Classification Journal of Library Automation, 2(1), 20(1969)

Dennis,64
Dennis,S., "Construction of a Thesaurus Automatically from a Sample of Text", Statistical Association Methods for Mechanized Documentation Symposium Proceedings, National Bureau of Standards Miscellaneous Publication 269, 1964, p113

Dewey, 65
Dewey Decimal Classification and Relative Index, 17th Edition, Forest Press, Inc., Lake Placid Club, New York, 1965

Dolby,68
Dolby, J., "The Distribution of Structure-Word-Free Back-Of-The-Book Entries", Proceedings of ASIS, 5, 65(1968)

Doyle,65
Doyle,L., "Is Automatic Classification a Reasonable Application of Statistical Analysis of Text?", Journal of the Association for Computing Machinery, 12(4), 473(1965)

Fischer,66
Fischer,M., "The KWIC Index Concept: A Retrospective View", American Documentation, 17(1), 57(1966)

Garfield,55
Garfield,E., "The Preparation of Printed Indexes by Automatic Punch-Card Techniques", American Documentation, 6, 68(1955)

Giuliano,65
Giuliano,V., "Interpretation of Word Association", Statistical Association Methods for Mechanized Documentation Symposium Proceedings, National Bureau of Standards Miscellaneous Publication 269, 1965, p25

Herner,62
Herner,S., "Methods of Organizing Information for Storage and Searching", American Documentation, 13, 3(1962)

Highcock,68
Highcock,S., "Natural Language Indexing for Automated Information Systems", in Classification for Information Retrieval edited by K. Bakewell, Archon Books, London, England, 1968, p65

Hines,70
Hines,T., and Harris, J., "Permuted Title Indexes: Neglected Considerations", Journal of the American Society for Information Science, 21(5), 369(1970)

Janaske,62
Janaske,P., "Manual Preparation of a Permuted-Title Index" BSCP Communique, Philadelphia,Pa, June, 1962

JCED,70
Beaton, R., Cameron, J., Lay, W., and Petrarca, A., editors, Author and Subject Index to Journal of Chemical and Engineering Data, 15(4) 600(1970)

Johnson,59
Johnson, A., "Experience in the Use of Unit Concept Coordinate Indexing to Technical Reports", Journal of Documentation, 19(3), 146(1959)

Johnson,68
Johnson,A., "Coordinate Indexing - A Practical Approach", in Classification for Information Retrieval edited by K. Bakewell, Archon Books, London, England, 1968, p73

Jordan,68
Jordan, J.and Watkins, W., "KWOC Index as an Automatic By-Product of SDI", Proceedings ASIS, 5, 211(1968)

Kennedy,63
Kennedy,R., "Writing Informative Titles for Technical Papers - A Guide to Authors", in Automation and Scientific Communication edited by H. Luhn, 1963, p133

Landry,69
Landry,B., "An Indexing and Re-indexing Simulation Model", Computer and Information Science Research Center Report 69-14, The Ohio State University, Columbus, Ohio, 1969

Lay, 70
Lay, W. and Petrarca, A., "Modified Double-KWIC Coordinate Index. Refinements in Main Term and Subordinate Term Selection", Social Impact of Information Retrieval, (Proceedings of the 7th Annual National Information Retrieval Colloquium), edited by A. D. Berton, Medical Documentation Service, The College of Physicians of Philadelphia, 1970, p155

Lejnieks,67
Lejnieks,V., "The System of English Suffixes", Linguistics, 29(2), 73(1967)

Lesk,66
Lesk,M., "Word Stem Terminators", in Information Storage and Retrieval, Scientific Report ISR-11 to the National Science Foundation, Department of Computer Science, Cornell University, Ithaca, June, 1966

Lesk,69
Lesk, M., "Word-Word Associations in Document Retrieval Systems", American Documentation, 20(1), 27(1969)

Lovins,68
Lovins,B., "Development of a Stemming Algorithm", Project INTREX, ESL-TM-353, Information Processing Group, Massachusetts Institute of Technology, Cambridge, Massachusetts, June, 1968, also in Mechanical Translation, 11(2), 57(1970)

Luhn,59
Luhn,H., "Keyword-In-Context Index for Technical Literature (KWIC Index)", RC-127, IBM Corp., Yorktown Heights, N.Y., 1959; also, American Documentation, 11(4), 288(1960)

Maizell,60
Maizell,R., "Value of Titles for Indexing Purposes", American Documentation, 11, 127(1960)

NAPS,69
NAPS Document NAPS-00682 from ASIS National Auxiliary Publishing Service, c/o Information Sciences, Inc., 22 West 34th St., New York, N.Y., 10001; remit $1.00 for microfiche or $3.00 for photocopies

Olney,63
Olney,J., "Library Cataloging and Classification", Report TM-1192, April, 1963,Systems Development Corporation, Santa Monica, California

Petrarca,69a
Petrarca, A. and Lay, W., "The Double-KWIC Coordinate Index. A New Approach for Preparation of High-Quality Indexes by Automated Indexing Techniques", J. Chem. Doc., 9, 256(1969)

Petrarca,69b
Petrarca, A. and Lay, W., "The Double-KWIC Coordinate Index II. Use of an Automatically Generated Authority List to Eliminate Scattering Caused by Some Singular and Plural Main Index Terms", ASIS Proceedings, 6, 277(1969)

Rosenberg,68
Rosenberg, K. And Bloeher, C., "A Comparison of Relevance of KWIC Versus Descriptor Indexing Terms", American Documentation, 19(1), 27(1968)

Ruhl,64
Ruhl, M., "Chemical Documents and Their Titles: Human Concept Indexing Versus KWIC-Machine Indexing", American Documentation, 15(2), 136(1964)

Salton,68a
Salton, G., "Use of Standardized Documentary Data in Automatic Information Retrieval", IEEE Transactions on Engineering Writing and Speech, 11(2), 101(1968)

Salton,68b
Salton, G., Automatic Information Organization and Retrieval, McGraw-Hill Co., N.Y, 1968

Salton,69
Salton, G., "A Comparison Between Manual and Automatic Indexing Methods", American Documentation, 20(1), 51(1969)

Sharp,66
Sharp,J., "The SLIC Index", American Documentation 17(1), 41(1966)

Simmons,63
Simmons, R. and McConlogue, K., "Maximum-Depth Indexing for Computer Retrieval of English Language Data", American Documentation, 14, 68(1963)

Skolnik,70
Skolnik, H., "The MULTITERM Index - A New Concept in Information Storage and Retrieval", Journal of Chemical Documentation, 10(2), 81(1970)

Stevens,66
Stevens,M., "Automatic Indexing: A State-of-the-Art Report", National Bureau of Standards Monograph 91, March, 1965

Taube,61
Taube,M., "Notes on the use of Roles and Links in Coordinate Indexing", American Documentation, 12(2), 98(1961)

Tocatlian,70
Tocatlian, J., "Are Titles of Chemical Papers Becoming More Informative?", Journal of the American Society for Information Science, 21(5), 345(1970)

Tukey,68
Tukey,J., "Multilingual Tail-Cropping", Report S-68-12, Department of Statistics, Princeton University, June, 1968

Vickery,68
Vickery, B., On Retrieval Systems Theory, Archon Books, London, England, 1968

Young,72
Young, C., "Design and Implementation of Language Analysis Procedures With Applications to Automatic Indexing", Ph.D. Dissertation, in Progress, Dept. Of Computer and Information Science, The Ohio State University

Zipf,49
Zipf, G., Human Behavior and the Principle of Least Effort, Addison-Wesley Publishing Co., Cambridge, Massachusetts, 1949

GLOSSARY

Abbreviations

AMT acutal main term

ASE actual subordinate entry

DKWIC double-KWIC

KWIC key-word-in-context

KWOC key-word-out-of-context

MMT maximal main term

PMT potential main term

PSE potential subordinate entry

SLIC selected listing in combination

Definitions

descriptor - a word or phrase describing a single concept

term - a combination of descriptors which describe a related collection of concepts

entry - a term and a means of locating a document containing the concepts described by the term

Notation

d<j> the jth document descriptor

i<j> the jth index descriptor

{k<1>,k<2>,...,k<n>} a set of n descriptors

(i=1,n) SUM (f(i)) the summation over i of the function f having argument i

k<i> UNION k<j> the union of elements or sets k<i> and k<j>

k<i> INTERSECT k<j> the intersection of the elements or sets k<i> and k<j>

INDEX

INDEX

The following is a KWIC-DKWIC index of this thesis prepared from the Table of Contents, List of Figures, and List of Tables. The numeric accession codes indicate the page on which the section heading or caption may be found. Captions are distinguished from section headings by the terminating letter P placed on the caption accession codes.

The index was generated by the KWIC-DKWIC subsystem described in appendix C, section C.5. Below are listed the index generation parameters and pertinent statistics for the index to follow.

```
  142 phrases
 1364 words
  120 primary stoplist words
  214 secondary stoplist words
  524 primary stoplist words found in titles
  759 secondary stoplist words found in titles
  605 distinct MMTs
  491 specificity 1 MMTs
  100 specificity 2MMTs
   12 specificity 3 MMTs
  143 distinct PMT groups
    9 maximum posting threshold
    9 minimum posting threshold
    9 permutation threshold
 1.21 average PMT specificity
```

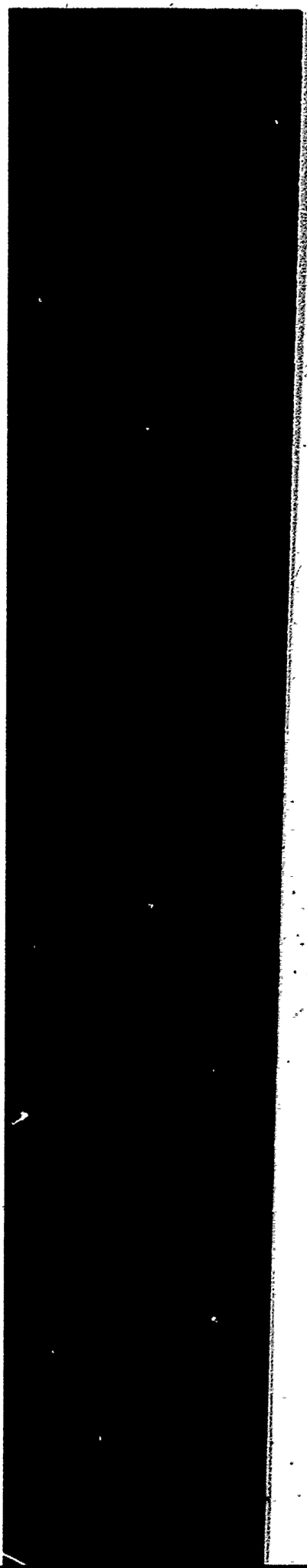
DKWIC INDEX(ES) (CONT)

INDEX(ES) (CONT)

INDEX(ES) (CONT)

INDEX(ES) (CCNT)

SELECTION(S) (CONT)



COMPUTER &
INFORMATION
SCIENCE
RESEARCH CENTER