solved by applying the fact that the atoms which formed the ring still have to form a chain.

Up to now only a limited number of reactions was encoded by the program. It is expected that more inconsistency checks and guide lines may be necessary when more reactions are processed. The program encodes only as deep as the description of the classification.[1] A greater depth of encoding will most likely not give rise to serious problems.

ACKNOWLEDGMENT

The authors wish to express their gratitude to the management of Gist-Brocades Research for the use of their IBM-1130.

REFERENCES AND NOTES

(1) M. Osinga and A. A. Verrijn Stuart, "Documentation of Chemical Reactions. I. A Faceted Classification", *J. Chem. Doc.*, **13**, 36–39 (1973).

(2) M. Osinga and A. A. Verrijn Stuart, "Documentation of Chemical Reactions. II. Analysis of the Wiswesser Line Notation", *J. Chem. Doc.*, **14**, 194–8 (1974).

(3) J. B. Hendrickson, "A Systematic Characterization of Structures and Reactions for Use in Organic Synthesis", *J. Am. Chem. Soc.*, **93**, 6847–54 (1971).

(4) Letter to W. J. Wiswesser, October, 8, 1974.

(5) M. F. Lynch, "Development and Assessment of an Automatic System for Analysing Chemical Reactions," Final report to the British Library, Research and Development, July 1975.

Computer Recognition and Segmentation of Chemically Significant Words for KWIC Indexing†

DAVID R. HEYM, HERBERT SIEGEL, MARGARITE C. STEENSLAND,* and HAO V. VO

Chemical Abstracts Service, The Ohio State University, Columbus, Ohio 43210

Received April 2, 1976

This paper describes an algorithm which recognizes and segments chemically significant words and thus provides additional index entry points for Keyword-In-Context (KWIC) Indexing. The words are those appearing in titles of documents selected for coverage in *Chemical Titles*. The procedure begins by matching strings of characters in the title word with roots, or stems, of words stored in computer memory. When a match occurs, parts of the recognized words are compared with other fragment lists, and a matrix is formed to select the fragments to be indexed. For example, dichloronitrobenzene is segmented as *di chloro nitro benzene*. Types of words recognized and segmented include names and classes of chemical substances, complex substituents, and reaction processes. The technique can be extended to words in other fields of science by appropriate modification of the lists and dictionaries stored in the computer. Implementation of machine segmentation improves consistency in the index production and eliminates most of the intellectual effort involved in the process.

INTRODUCTION

Chemical Titles (CT) is a biweekly current-awareness service which reports the titles and bibliographic information of those documents of chemical and chemical engineering interest which have recently been published in selected journals. The production of CT is computer-based and results in a machine-readable version as well as the printed issue.

Although the printed issue of CT contains a bibliographic section and an author index, the basic access tool for the information is the Keyword-In-Context (KWIC) Index (see Figure 1). Each line of the index contains up to 60 characters and spaces, plus a code identifying the specific article in a journal.

The access point for each line is the keyword (lipid, lipids, etc., in Figure 1). Keywords appear in their normal context in the title and are listed alphabetically in a fixed, columnar arrangement.

Each significant word in the title is used as the keyword or entry point for a new index line. A significant word is defined to be any technical term describing a process, chemical substance name, class of chemical substances, animal name, organ, chemical reaction, operation, apparatus or equipment, theory or hypothesis, or scientific law. A nonsignificant word is one which is not in one of the above classes or which is too general for index purposes. The "CT Stopword List", printed in each issue of CT, alerts users to nonsignificant words.

Words are frequently made up of several fragments or segments, some of which are themselves words that have meaning and therefore have value as access points in a KWIC Index. When this is true of chemically significant words (chemical names, chemical processes) in the titles processed for the KWIC Index in CT, those words are segmented to yield additional index entries. In Figure 1, the next to last entry is indexed at the segment *lipo* of the keyword *apolipoprotein*. Later in the KWIC, the same entry is indexed at the segment *protein*.

The object of the work described in this paper was to develop a computer-based algorithm which would provide segmentation for indexing purposes, i.e., find and indicate access points within chemically significant words. Since segments derive their significance from their technical meaning, not all are suitable index terms. Examples of acceptable segmentation are shown in Figure 2; the apostrophes indicate the segmentation points.

The segmentation algorithm must recognize chemically significant words, distinguish between general terms and chemical names, and indicate proper segmentation points. It must also recognize words which are candidates for segmentation but which cannot always be segmented because of ambiguity. These words must be listed for review and proper handling by chemists.

The advantages of such an algorithm include elimination of much of the intellectual effort otherwise required and an increased consistency in the keyword index. In the Chemical

† Presented before the Division of Chemical Information, 170th National Meeting of the American Chemical Society, Chicago, Ill., Aug 27, 1975.

* Author to whom correspondence should be addressed.

```
otential generation in bilayer lipid membranes in the NADH-flavine BBACAQ-0368-0022
tra of spin labels in phospho lipid membranes.= +etic resonance spec JOMRA4-0016-0001
nt on liver lipid content and lipid metabolism of rats fed a non prote JNSVA5-0020-0215
osslinking of amino phospho lipid neighbors in the erythrocyte memb BBRCA9-0061-0030
he submicrosomal membrane lipid of rat brain.= +r composition of t JCLBA3-0063-0749
r microsomes subsequent to lipid per oxidation induced destruction o BBRCA9-0061-0258
             Present and future lipid studies.=                     KASEAA-0012-0602
a carbon-14-labeled manno lipid with properties of dolichyl manno JBCHA3-0249-6372
xtraction of liver microsomal lipid. II. Effect on+ Organic solvent e MOPMA3-0010-0963
                  Synthesis of lipid-soluble vitamins.=             YKGKAM-0023-0633
ation of glucose carbons into lipids by isolated rat lung.= +incorpor AJPHAP-0227-1103
ements of lateral diffusion of lipids in biological membranes.= +asur CHPLBC-0028-0280
neticall+ Concentrations of lipids in the adrenal cortical tissue of ge AJBSAM-0027-0651
nd abnormal accumulation of lipids in the liver+ Intake of sucrose a KASEAA-0012-0598
membr+ Newly synthesized lipids incorporated into influenza virus NATUAS-0251-0333
             composition of the lipids of coccidioid fungus.= Chemical MZUKAV-0036-0595
of the interaction of phospho lipids with synthetic poly peptidies by t BIOHAO-0039-1003
             New anionic glyco lipids.=                             SEIKAQ-0046-0771
tics in relation to sex, serum lipids, body weights, and gallbladder dis JCINAO-0054-1301
ose in the filamentous blu+ Lipo poly saccharide containing L-acofri JOBAAY-0120-0672
cycline on ultrastructure and lipo protein secretion in the rat hepatoc TXAPA9-0030-0317
e bilamellar vesicles and apo lipo protein-alanine (APO-C-III).= +in BICHAW-0013-4765
mones and cortico sterone on lipolysis in hypophysectomized rats.= ACENA7-0077-0502
```

Figure 1. CT KWIC Index.

- Substance names
  - anthra'quinone
  - methyl'benzyl'ammonium
  - meth'acrylic
  - di'fluoro'benzene
  - mono'phosphate
  - cyclo'pentadienyl
- Process names
  - meth'oxy'carbonylation
  - co'polymerization

Figure 2. Segmentation of chemically significant words.

Abstracts Service (CAS) bibliographic data handling system, the computer algorithm also permits integration of the workflow, since it eliminates a special input operation for selected titles.

RESEARCH METHODOLOGY

The CAS staff had previously developed a program[1] that distinguishes names of chemical substances from other words, and this program became the basis for the first phase of the computer algorithm. Some of the techniques employed in a systematic nomenclature searching program[2] were helpful in programming the second phase.

None of the other literature proved helpful in achieving the objectives of the algorithm. For the most part, the design of the second phase evolved from a study of the segmentation process as performed by chemists/editors applying their technical knowledge and expertise.

The manual recognition and segmentation process was basically routine, but coping with the irregularities and unusual cases required highly trained technical personnel. The nonroutine work accounted for approximately one-third of the total effort.

RATIONALE FOR THE ALGORITHM

To construct the algorithm, it was useful to examine the process used by editors performing the segmentation function. The analysis revealed that the editors (1) realize the purpose of the segmentation; (2) know a set of words to be segmented and rules for determining set membership; (3) classify new or unfamiliar words as members or nonmembers of the given sets; (4) realize the meanings implied in adjacent letter/character strings in words in the sets; (5) infer the meanings of relatively unfamiliar letter/character strings; (6) insert segmentation points in a predictable and reproducible manner.

Point 1 above is addressed indirectly in the algorithm, since that function is intrinsic to the testing and processing of the words. Points 2 and 3 are addressed in the first of the two phases of the algorithm, the recognition scheme. The recognition scheme selects words for, or rejects them from, processing by the second phase. Once a word has been classified as a candidate for segmentation, the segmentation scheme addresses points 4, 5, and 6 in the analysis.

Test	Yes	No
1. Length ⟨5?	Reject	To Test 2
2. Length =5?	To Test 3	To Test 4
3. Special (aroxy, azoxy, dioxo, dioxy, epoxy)?	Recognize	Reject
4. Stopword?	Reject	To Test 5
5. Dictionary match?	To Test 6	Reject
6. Negative term?	Reject	Recognize

Figure 3. Recognition tests.

RECOGNITION SCHEME

Figure 3 shows the tests made during the recognition scheme. Reject means that no further tests will be imposed (i.e., the word is *not* a candidate for segmentation); recognize means that the word *is* a candidate for segmentation; stopwords are those which are not used for indexing in the CT KWIC Index.

In the first test, a word of less than five alphabetic characters is rejected because such words require no segmentation within the rules to be applied. The second and third tests recognize the only five-letter words which require segmentation. Test 4 screens out stopwords.

The last two tests utilize a dictionary of approximately 1600 terms (letter/character strings) which occur in chemically significant words. Most of the strings resemble truncated search terms. A negative term is a string of letters characteristic of chemically nonsignificant words with imbedded shorter strings which are characteristic of chemically significant words. If words containing negative terms were recognized and processed further, undesirable segmentation could result. The search technique used rejects or recognizes the word based on the longest character string in the word found to match a dictionary term. For example, "chloroplast" is a negative term; therefore, the word will be rejected even though "chloro" is chemically significant.

SEGMENTATION SCHEME

Recognized words advance to the segmentation scheme. Its operation is based upon the fact that most chemically significant words are constructed from segments or letter/character strings according to the rules or patterns of chemical nomenclature. One particularly common pattern has the form: multiplier [mono, di, tri, etc.] + substituent [chloro, fluoro, bromo, cyano, etc.] + parent compound [phenol, hexane, etc.].

The segmentation scheme consists of three routines:

1. Pattern Filling. The word is translated into a number matrix for subsequent processing.

2. Pattern Matching. The pattern in the first matrix is compared against a set of patterns, and, when a match occurs, a second matrix is filled. These patterns are formal descriptions of the syntax of chemical nomenclature.

3. Resolution/Checking. The number content of the second matrix is analyzed; nonsignificant numbers are dropped; all remaining numbers are translated into segmentation markers.

Pattern Filling. The pattern-filling subroutine analyzes an input word by generically describing its parts, e.g., *di*, in dimethyl, is a multiplier. This subroutine uses a list, or dictionary, of approximately 1700 character strings which have been generically classified and assigned three-digit type numbers, or codes, which represent the various grammatical and chemical functions of the strings. The list is ordered and searched in such a way that the number linked to the longest

Table I. Classification Scheme for Letter Strings

Type no.	Function/limitations	Examples
021	Multiplier-not *tris*	di, tri, tetra
022	Multiplier-not *bis*	bi, uni, deci
023	Multiplier-*bis* only	
024	Multiplier-*tris* only	
103	Multiplier-not ending with *a*	hex, pent, dec
104	Multiplier-type 021, *tri* through *octa*, immediately followed by *dec*	tridec
031	Suffix-terminal	acin, mycin
046	Suffix-heterocycle	epane, onane
048	Suffix-process endings	ylation
052	Suffix-systematic/pseudo systematic	ol, yl
053	Suffix-systematic/pseudo systematic; ending with *i* or *o*	alo, amo
054	Suffix-systematic/pseudo systematic; beginning with *an, in,* or *on*	onate
055	Suffix-systematic/pseudo systematic; with both limitations above	ano, ino
102	Suffix-not adjacent to types 052-055	eic
041	Prefixes/Infixes-may be preceded by multipliers	oligo
042	Prefixes/Infixes-never preceded by multipliers	sub, pyo
043	Prefixes/Infixes occurring in heterocyclic nomenclature	cyclo
044	Prefixes/Infixes-heteroatom infixes, short form	stann
045	Prefixes/Infixes-heteroatom infixes, longer form	stanna
081	Prefixes/Infixes-short form of type 082	aut
082	Prefixes/Infixes-same as type 042, but the last letter may belong with the following string	auto
083	Prefixes/Infixes-short form of type 084	hydr
084	Prefixes/Infixes-same as type 041 but the last letter may belong with the following string	hydro
135	Prefixes/Infixes-five-letter specials and five-letter strings requiring the same segmentation	See Figure 2; also didec
136	Prefixes/Infixes-strings longer than five letters, requiring the same type segmentation as type 135	methoxy
071	Alkaloids ending in amine and related words	cethiamine
072	Class names and words requiring internal segmentation	polymer glycolipid
101	Element names-significant parts only	iron, tin
121	String of length 1-otherwise unclassified	
122	String of length 2-otherwise unclassified	
123	String of length 3-otherwise unclassified	
134	String of length ≥4-otherwise unclassified	
141	Locant/nonletter string without a hyphen	N.A.[a]
151	Locant string with one or more hyphens	. . .-1,3-. . .
161	Locant/nonletter string without a hyphen, with brackets or parentheses	. . .[2.2.2]. . .
171	Locant/nonletter string with one or more hyphens and brackets or parentheses	. . .-(2)-. . .

[a] In practice, this is a temporary assignment until one of the other three possibilities is identified.

TRIHALOBENZENES				DICHLOROBENZENES			
	T	*F*	*L*		*T*	*F*	*L*
(TRI)	021	1	3	(DI)	021	1	2
(HALO)	084	4	7	(CHLORO)	084	3	8
(BENZ)	082	8	11	(BENZ)	082	9	12
(ENE)	052	12	14	(ENE)	052	13	15
(S)	121	15	15	(S)	121	16	16

Figure 4. Pattern-filling results.

string which matches a string in the input word will be retrieved. Thirty-two different type numbers represent the functions of letter strings, and four type numbers represent the functions of various nonletter strings and letter/character strings in which the length of the alphabetic component is less than five. The generic classification scheme used is given in Table I.

The pattern-filling routine constructs a table of numbers, three columns wide, which completely describes the word in terms of its internal strings. Two typical tables are shown in Figure 4. None of the alphabetic information is part of the table; it is included only to show the strings and the words being analyzed.

There is one row in the table for each string in the word that matches a term on the list. The leftmost column (T) gives the type number retrieved from the list, the second column (F) gives the position within the word of the first letter/character in the string, and the last column (L) gives the position within the word of the last letter/character in the string. The table is finished when the number in the third column of a row is equal to the number of letters in the word.

Word	Questioned Letter	String	Table Entry		
pentaene	a	penta	021	1	5
pentane	a	pent	103	1	4
trisoxazole	s	tris	024	1	4
trisaccharide	s	tri	021	1	3
acrylonitrile	o	ylo	053	4	6
phenyloxazole	o	yl	052	5	6

Figure 5. VOWELEND string differentiation.

The pattern-filling routine contains several subroutines which are, in effect, refinements that differentiate the environments of certain letter strings, i.e., strings with specified type numbers.

If the type number retrieved for the string under consideration indicates that the string is a multiplier and if the string is followed by one or more multipliers (e.g., dotriacontahectane), the first refinement will permit the stringing together or concatenation of the multipliers, regardless of their order. This makes it unnecessary to include all possible permutations of multipliers on the list.

The second refinement deals with cases where the longest match is not necessarily the best candidate for segmentation. It determines whether the last letter in the string under consideration should remain a part of that string or whether it should be the first letter of the subsequent string. It is applied when the type number retrieved for the original string is 021, 023, 024, 045, 053, 055, 083, or 084. Since the letter in question is usually a vowel, the routine is called VOWELEND. Its application is illustrated in Figure 5.

The VOWELEND routine determines to which string the questioned letter belongs by attempting to match the string which follows the letter with an entry on the list. If it finds a match which is not a suffix, the original string is accepted. For example, both *tris* and *tri* appear on the list used by the pattern filling routine. After finding the string *tris* in a word and retrieving its type number (024), the VOWELEND routine attempts to find a match on the list for the character string following *tris*. In the word trisoxazole, *oxa* is a term on the list; therefore the original string *tris* is retained, and the corresponding table entry shown in Figure 5 results. However, there is no match for the string *accharide* in the word trisaccharide. Therefore the final letter of the original string is affixed, and the routine searches for a match on *saccharide*. It is successful in this attempt; therefore the original string is shortened to *tri* and the entry shown in Figure 5 will be the first row in the table for that word. If only one search results in a match, and the string is a suffix, then the suffix is used; if both result in matches and both are suffixes, the longer suffix is used. If both attempts result in no matches, the original match is presumed correct, and the appropriate type number (121, 122, 123, or 134) for unclassified strings will be filled into the next row.

There are two other refinements in pattern filling. In one, if the type number of the matching string is one of the special types, 135 or 136, the subroutine will divide the original string into two strings, the second of which has three characters. Then it will fill in two rows in the table. It will enter the type number 135 or 136 in the first row, along with positional numbers for the first part of the string. It will put the same type number into the next row, along with positional numbers for the last three letters of the original string. The last refinement also divides a string into two parts. The string in question must have the type number 072 and must have a type 072 string embedded at the end of the string (e.g., *glycolipid*, type number 072; *lipid*, type number 072). If this is the case, two entries will be made in the table representing the word. This refinement is similar to VOWELEND in effect and is somewhat more reliable for nonsystematic substance names.

Pattern Matching. After the pattern-filling routine has completed the analysis of a recognized word, the pattern-matching routine uses the completed table to build a weight factor matrix, which is subsequently used by the last routine to determine the satisfactory segmentation of the word.

This routine recognizes more than 50 patterns, where a pattern expresses the mandatory, negated, or optional adjacency of two given type numbers in the table in much the same way that nomenclature rules dictate word segment proximities. Each pattern is associated with one of twenty different responses which direct the routine in building a two-column matrix with one more row than there are letters in the word. Table II lists just a few of the patterns and responses used in the pattern matching routine. Matching the patterns in the tables in Figure 4 against the patterns in Table II and carrying out the indicated responses yield the weight factor matrices in Figure 6.

The nonzero numbers in the matrices are the weight factors. The single-digit numbers in the rightmost column of each matrix are also called segmentation markers. They indicate potential segmentation points behind the position they hold in the matrix. The two-digit numbers in the left column of each matrix indicate points at which a test should be made to validate placement of the segmentation markers.

	0	1
(T)	0	0
(R)	0	0
(I)	0	1
(H)	0	0
(A)	0	0
(L)	0	0
(O)	0	1
(B)	0	0
(E)	15	0
(N)	0	0
(Z)	0	1
(E)	0	0
(N)	0	0
(E)	0	2
(S)	11	0

	0	1
(D)	0	0
(I)	0	1
(C)	0	0
(H)	0	0
(L)	0	0
(O)	0	0
(R)	0	0
(O)	0	1
(B)	0	0
(E)	15	0
(N)	0	0
(Z)	0	1
(E)	0	0
(N)	0	0
(E)	0	2
(S)	11	0

Figure 6. Pattern-matching results.

Table II. Pattern Responses

Pattern		Response	
Type no. of current string	Followed by type no.		
1. 21 or 22 or 23 or 24	82 or 84	1a.	Put a weight (1) before the first character of current string.
		1b.	Resume matching with string following current string.
2. 83 or 84	any	2a.	Put a weight (1) ahead of and at the end of the current string.
		2b.	Resume with string following current string.
3. 81 or 82	≠102	3.	Same response as 2a and 2b.
4. 52 or 54	any	4a.	Put a prevent weight (15) three characters ahead of current string
		4b.	Put a weight (2) at the end of the current string and resume with string following the current string.
5. 121	any	5.	Resume matching with string following current string.
6. [End of table]		6a.	Put a terminal weight (11) at the last character.
		6b.	Proceed to the resolution routine.

The pattern-matching routine detects certain error conditions in words which cause processing by the algorithm to cease and cause the words to be printed out for handling by chemists. This occurs if the word contains (1) a multiplier followed by an unlisted string; (2) a terminal suffix which is not at the end of the word; (3) several adjacent suffixes, the last of which ends in *o* or *i* and is followed by an unlisted string; (4) a compound multiplier (type 104) followed by the suffix *yl*; (5) an unlisted string followed by an element name.

Resolution/Checking. This routine uses the weight factor matrix produced by the previous routine to perform the actual segmentation of the chemically significant words in the title.

The first thing that it does in the process is to remove invalid segmentation markers, markers at points where segmentation is undesirable. For example, in the weight factor matrices shown in Figure 6, the 1 in the position of the letter Z would be removed from each matrix, because each falls between a weight of 15 and a weight of 2. This indicates that it is a weight immediately preceding a suffix. The suffix is not to be used as an access point in the KWIC Index.

There are six tests for the validity of a segmentation marker. Their effect is to remove markers at the beginning and at the end of a word, to remove a marker before a suffix, and to remove markers adjacent to locants or internal punctuation. The latter are removed because the KWIC Index generation programs automatically provide such breakpoints. These tests cancel adverse segmentation which might result from string analysis alone.

After undesirable markers are removed, three other tests are performed before the word is finally segmented. Placement of segmentation points is considered impossible, by definition, if (1) two markers are adjacent in the table; (2) a marker precedes a two- or three-letter string, and that string precedes a type 072 string; (3) a marker precedes a string which begins with certain two-letter strings (such as nt, rt, cb, etc.).

When one of the above error conditions exists, the word will be printed out for a chemist's review. If there is no error condition and if a segmentation marker remains in the weight factor matrix, a segmentation point (nonprinting character) will be interposed following each remaining marker.

DESIGN CONDITIONS

The feasibility of the entire procedure depends on the capability of matching or not matching the candidate strings with the strings in the dictionaries in the fewest number of comparisons. Three means were used to meet this objective: advantageous arrangement of the terms on the lists, hash codes, and binary searching.

Because of the static nature of the recognition dictionary, it was possible to order the entries according to the first two letters of each word fragment. As the program traverses a word from left to right, it uses adjacent letter pairs to compare dictionary entries for a match. A match occurs when the dictionary entry is fully contained within the word under investigation. Although a word may contain more than one segment that can be matched, it is the first match that triggers the response, and thus that is the only one seen in the recognition phase. For this reason, negative terms precede positive terms, and such a match causes the program to proceed to the next word. Other features of the arrangement are: (1) when there are no entries for a letter pair, the search terminates; (2) related entries are stored in adjacent locations; (3) the typical matched entry may require less than ten comparisons (cf. a binary search on a table of 600 entries which averages about ten comparisons[3]); (4) when there are many segments beginning with the same two letters, 40 or 50 comparisons may be required.

In the segmentation phase of the program, the aim is to find the longest string on the dictionary which matches a string in the word. Therefore, the 1700 strings in the list are arranged by length with the longest strings first, and alphabetically within the length. A hash code[4] is computed from the sum of the values of all two-letter pairs in each string. In investigating an input word, the program first generates its hash code, then compares it with the hash codes of equal length strings on the list. If none of the hash codes agree or if there are no strings of that length on the list, there is no chance for a match. The last letter of the word is dropped, a new hash code is computed, and the process is repeated. When the hash codes do agree, a binary search is performed on the dictionary entries which have the same length as the input string to be matched.

RESULTS

The production program processes more than 6000 titles per issue. The average title has more than twelve words, two of which are candidates for segmentation. For development purposes, the procedure was programmed in a high-level programming language, PL/1, for an IBM 360/370 series computer. After the design was complete and the results were satisfactory, it was rewritten in Basic Assembler Language (BAL) to achieve maximum efficiency in execution. The finalized BAL procedure uses approximately 65K bytes of core memory for program instructions, dictionaries, buffers, and dynamically allocated work areas. Processing a complete journal title requires less than 15 msec of execution time. The CAS data base management system[5] is used, as are several in-house software modules.

Sample: 9,726 titles; 108,891 words

● Recognized	15,765
Errors	50
● Printed for review	155
● Segmented	15,610
Errors	337
Recognition efficiency	99.9%
Segmentation efficiency	98.1%

Figure 7. Summary of results.

The system using the algorithm was installed into the bibliographic headings input system in July 1974, and an analysis was conducted on the segmentation results over the first 20 days of input. The analysis is summarized in Figure 7.

DEFICIENCIES AND FAILURES

There are two difficulties with the algorithm. One is the unpredictable segmentation of hyphenated words in which the hyphens are not a part of a locant chain (as in "carbon-modified") and in which only one member of the compound word is recognized. Some such troublesome words have been identified and placed on the list with the type number 134 and are thus ignored.

The other difficulty is the inconsistent segmentation of names of heterocyclic compounds. When the algorithm was devised, it was decided that segmentation would not be introduced into the parent name of these compounds if the name followed heterocyclic nomenclature conventions. Unfortunately, certain letter strings (ox, thi, phosph, az) occur in both heterocyclic and substituent names, but segmentation is desired only in the latter case. Since it was not desirable to include letter strings having two different classification numbers, the algorithm leans toward over segmentation.

HUMAN INTERFACE

Words which the algorithm cannot segment because of an error detected in the word are printed on a report which is directed to a chemist assigned to perform the corrections. The desired segmentations are marked on the report and keyed into the computer system.

Consistent segmentation by the program depends upon the machine-readable dictionaries, which require periodic updating. The same group of chemists who handle the problem words is also responsible for maintaining the dictionaries. To aid the staff in this function, a printed manual describing the algorithm and the dictionary philosophies has been prepared.

OTHER USES OF THE ALGORITHM

The segmentation algorithm has been slightly modified for use in conjunction with the IBM hyphenation routines and with a CAS-developed molecular formula hyphenation algorithm to detect logical hyphenation breakpoints in text. This composite hyphenation algorithm has been applied to computer-controlled page formatting of primary journal text.

In order to use the segmentation algorithm for this purpose, some of the constraints within the last part (Resolution/Checking) were removed. Those constraints suppressed the insertion of segmentation markers following a natural indexing point, such as a bracket or parentheses. For purposes of hyphenation, such segmentation markers are retained.

CONCLUSION

Computerized segmentation of title words for a KWIC Index yields large benefits. For CT, these benefits include reduction of technical staff time from 4000 to 750 hr per year, reduction of typing paper and computer reports from 30 000 sheets to 8500 sheets, and reduction of computer execution time from 3 to 1 hr per year. In addition, the bibliographic data for CT are now provided from input for other CAS products. When chemists performed the segmentation process, the segmented titles were entered into the CAS data base for use in the CT KWIC Index, while the unsegmented titles were entered for use in other products. Thus one input process has been eliminated. This eliminates 7500 hr of typing and keyboarding per year.

The CAS word recognition and segmentation system is limited to chemically significant words because of the content of the dictionaries and the nature of the patterns. The system could be applied to many other disciplines by (1) selecting the types of words to include and exclude (this is the most difficult part of the work); (2) constructing a list of classified character strings in the recognized words (this would be similar to preparing a search profile with left and right truncation); (3) constructing a list of type numbers around which segmentation points should or should not fall; (4) using the applicable pattern-filling rules; (5) extracting applicable patterns and responses; (6) evaluating the applicability of the programs that remove undesirable segmentation markers.

ACKNOWLEDGMENT

CAS is pleased to acknowledge the funding of this work by the National Science Foundation (Contract C656).

LITERATURE CITED

(1) D. L. Dayton, D. R. Heym, R. Salvador, and G. M. Vanautryve, "CA Subject Index Vocabulary Standards (NSF C521 Task M and C656 III.1), Technique Description, Chemical Name Screen Algorithm", CAS Internal Report, 1972.

(2) Chemical Abstracts Service, " Substructure Searching of Computer-Readable CAS 9CI Chemical Nomenclature Files", ISBN 8412-0204-4, American Chemical Society, Washington, D.C., 1974.

(3) D. E. Knuth, "The Art of Computer Programming, Sorting and Searching", Vol. 3 Addison-Wesley, Reading, Mass., 1973, p 411.

(4) Reference 3, p 506 ff.

(5) Chemical Abstracts Service, "Facility for Integrated Data Organization (FIDO), User Reference Manual", NTIS, PB-236-020, Springfield, Va., 1974.

An International Mass Spectral Search System (MSSS). V. A Status Report

RACHELLE S. HELLER[†]

Computer Science Department, University College, University of Maryland, College Park, Maryland 20742

G. W. A. MILNE[‡] and RICHARD J. FELDMANN[‡]

National Institutes of Health, Bethesda, Maryland 20014

STEPHEN R. HELLER[*]

Environmental Protection Agency, MIDSD, PM-218, Washington, D.C. 20460

Received April 19, 1976

The status of MSSS is described. Problems and experiences that have been encountered in three years of commercial operation of this system are reported and discussed.

INTRODUCTION

This article has been prepared to present the experiences of the international Mass Spectral Search System (MSSS) which has been operating commercially for almost three years. The MSSS is a unique system in many ways, notably in that it is a cooperative venture between two governments (U.S. and U.K.). In addition, within the U.S. government, four agencies, EPA, NIH, FDA, and NBS, are also working together toward the same goal. This structure is not free of complications, but the problems that have arisen have, for the most part, been solved, and it is felt that international collaboration of this sort is both feasible and worthwhile.

BACKGROUND

In the period 1971–1972 both EPA and NIH began to develop computer systems for aiding in the identification of compounds from their low resolution mass spectra. Both groups began by using a data base prepared by the Mass Spectrometry Data Centre (MSDC) at Aldermaston, U.K. Following some limited attempts by both EPA and NIH to disseminate their systems to the scientific community, where there was considerable interest in the MSSS, a collaborative effort was established between them and the MSDC. Under this arrangement, EPA and NIH, later joined by FDA and NBS, are funding continued development of the MSSS, while the U.K. government takes responsibility for making the MSSS available to the international scientific community. In September 1973, the MSSS was thus made available on the GE Mark III computer network. In July 1975, for technical and economic reasons, the MSSS was transferred to the ADP-Cyphernetics International Computer network.

USAGE

At present over 125 separate organizations, involving about 175 laboratories in North America and Western Europe, are using the MSSS on a daily basis. From a start of about 10 searches per day in October 1973, the rate of use has grown to over 100 per day some two years later. In addition, about 35 reference spectra from the master file are retrieved (plotted or printed) every day. User interaction with the system developers is moderate, with an average of three "CRABS" (comments or complaints written by users onto the system disk file) per week. These CRABS consist of problems, requests

[†] University College.

[‡] National Institutes of Health.

[*] Environmental Protection Agency. Author to whom correspondence should be addressed.