On-Line Indexing Experiment at Chemical Abstracts Service: Algorithmic Generation of Articulated Index Entries from Natural Language Phrases†

KENNETH H. BASER, STANLEY M. COHEN,* DAVID L. DAYTON, and PENNELL B. WATKINS

Chemical Abstracts Service, P.O. Box 3012, Columbus, Ohio 43210

Received October 11, 1977

An experimental on-line indexing program developed at Chemical Abstracts Service generates articulated index entries from natural language phrases input by a document analyst. The analyst may allow the program to generate a list of candidate index entries from a particular input phrase or may specify that only certain words within the phrase be used as index headings. In either case, all the program-generated index entries are displayed on a CRT terminal for final selection by the analyst. During small-scale testing of the program, data on human factors, index entry quality, and analyst work rates were collected. These data suggest that the technique is feasible for use in production.

INTRODUCTION

The Ninth Collective Index presently being compiled by Chemical Abstracts Service (CAS) will consist of over 100000 pages and will occupy 62 volumes when completed in 1978. It will contain almost 20 million index entries, including 7.3 million Chemical Substance entries and 3.7 million General Subject entries. Molecular Formula and Author Index entries make up the remaining 9 million index entries. The generation of these index entries is one component of the total document analysis task performed by CAS professional staff. Figure 1 displays a set of index entries which could be generated for a given document.

Each of the three index entries shown in Figure 1 consists of two components: (1) the primary access term of the index entry upon which the alphabetic arrangement of the index is based—called the "Heading"; (2) the indented subordinate modifying phrase—called the "Text Modification". In the present CAS Index Production system, each index entry is individually dictated by a document analyst into a tape recorder and is subsequently keyboarded by clerical staff for input to the computer-based system.

The level of effort of both professional and clerical staff involved in the generation and editing of the *Chemical Abstracts* (CA) Chemical Substance Index and General Subject Index is substantial. CAS is, therefore, interested in exploring ways of improving the efficiency of the input and editing operations while maintaining or improving the quality of *Chemical Abstracts* indexes.

Armitage and Lynch[1,2] have described a process called phrase articulation in which multiple index entries of the type published in the CA Chemical Substance Index and General Subject Index are generated from a single natural language phrase by means of an algorithm. Several versions of the articulation algorithm existed at various times during the course of its development. Some early work on the algorithm was carried out by Salvador.[3] A later version of the algorithm was described by Cohen et al.[4] Figure 2 illustrates the phrase articulation process utilizing the version of the articulation algorithm as it existed at the conclusion of the developmental work being reported on in this paper. The single phrase shown in the top part of Figure 2 can be transformed by the articulation algorithm into the three index entries in the lower portion of the figure. In the work reported by Cohen et al.,[4] an experiment was conducted in which document analysts dictated phrases into tape recorders. The document analysts indicated those words in the phrase which were to be used as headings, by means of dictated codes. The dictated phrases were then processed in batch mode by the articulation algorithm and sets of index entries generated. While this earlier experiment indicated that index entries of acceptable quality could be generated by the algorithm in a high proportion of cases, a major problem was encountered with regard to the errors in the dictated codes. The dictation of the codes (which served several vital functions including delimiting headings within phrases) proved to be an error-prone process. Hall[5] encountered analogous difficulties with keyboarded flags used to designate headings within input phrases. The coding error level was considered too high to be tolerable in a production environment.

Interactive processing appeared to offer a solution to the flagging and coding problems in that the equivalent of the dictated codes and flags could be generated by an on-line program in response to a series of actions taken by analysts in designated fields on screens appearing on a CRT. It was expected that interactively applied programmed edits would greatly reduce or eliminate the coding inconsistencies which frequently occurred during the earlier batch-dictation phrase input experiment. Based on these expectations and on recent favorable experience with an on-line editing system at CAS,[6] it was decided to develop an experimental On-Line Phrase Input (OLPI) program and to test the interactive indexing method in a production-like environment. This paper describes the On-Line Phrase Input program and its testing.

DESCRIPTION OF ON-LINE PHRASE INPUT PROGRAM

The On-Line Phrase Input program was written in IBM assembler language and supports a single IBM 3277 terminal connected to an IBM 370/168 computer. In the course of the experimental work covered in this paper, the program underwent a series of modifications which included extension of program capabilities and optimization of the human-machine interaction. The description which follows portrays the program as it existed following all such modifications. The program is described from an operational viewpoint using a progression of diagrams of screens generated by the program in the processing of phrases to produce articulated index entries. In these screen diagrams, data keyed by the document analysts are shown enclosed in rectangles for illustrative purposes, to distinguish them from data generated by the program. The program does not actually display keyed data in this manner.

Function Keys. The terminal used in the experiment has 12 function keys which can be programmed to carry out specific operations. These function keys were utilized in the experimental On-Line Phrase Input program and significantly

†Presented before the Division of Chemical Information, 173rd National Meeting of the American Chemical Society, New Orleans, La., March 22, 1977.

* Author to whom correspondence should be addressed.

0095-2338/78/1618-0018$01.00/0

HEADING TEXT MODIFICATION

Sodium thiocyanate
glycoproteins secretion in tracheal mucus in relation to

Glycoproteins
secretion, in tracheal mucus, sodium thiocyanate effect on

Mucus
tracheal, glycoproteins secretion in, sodium thiocyanate effect on

Figure 1. Set of index entries for a document.

PHRASE

effect of sodium thiocyanate on glycoproteins secretion in tracheal mucus

Sodium thiocyanate
glycoproteins secretion in tracheal mucus in relation to

Glycoproteins
secretion, in tracheal mucus, sodium thiocyanate effect on

Mucus
tracheal, glycoproteins secretion in, sodium thiocyanate effect on

ARTICULATED INDEX ENTRIES GENERATED FROM PHRASE

Figure 2. Phrase articulation process.

ENTER PHRASE

76678546X

effect of sodium thiocyanate on glycoproteins secretion in tracheal mucus

Figure 3. ENTER PHRASE screen.

enhanced the general workability of the program. The 12 function keys were used to carry out specific operations, some of which will be discussed later. However, most of the normal forward processing of phrases can be carried out using a single key—the ENTER key.

Screen for Keying of Phrases. The screen used by the document analyst for keying the phrases is illustrated in Figure 3. The number in the upper right-hand portion of the screen is a document identification number (keyed on an earlier screen). The phrase is keyed by the document analyst on the ENTER PHRASE screen as illustrated in Figure 3.

Flagging Options. Following the keying of a phrase, the document analyst has two options available with respect to the manner in which portions of the phrase are designated as headings for the index entries he wishes to generate from the phrase.

1. Manual flagging: in this option, the document analyst manually designates those words in the phrase he wishes to use as headings of index entries. The program then generates a text modification for each heading selected.
2. Automatic flagging: in this option, the analyst allows the program to generate a list of candidate headings from which he may subsequently accept or reject individual index entries as desired.

We will first examine the manual flagging option. Once the phrase is keyed on the ENTER PHRASE screen, the document analyst selects the manual flagging option by hitting

FLAG HEADING

76678546X

effect of sodium thiocyanate on glycoproteins secretion in
1 2 3 4 5 6 7 8

tracheal mucus
9 10

HDG: 3,4

NUM: 1

CLASS:

Figure 4. FLAG HEADING screen.

the ENTER key. After the ENTER key is struck, the program displays a FLAG HEADING screen which is illustrated in Figure 4.

Flag Heading Screen. When the program displays a FLAG HEADING screen, the following data items appear:

1. The document identification number (upper right-hand portion of screen).
2. The phrase keyed on the ENTER phrase screen, displayed on the FLAG HEADING screen with each word individually numbered by the program.
3. A data field labeled with the mnemonic code HDG (for heading). The HDG field is used by the document analysts to designate headings within the phrase using the program-supplied numbers.
4. A data field labeled NUM. This data field is used to distinguish between Chemical Substance and General Subject index entries. In the CAS index production system, PAR is the mneumonic code used to designate index entries intended for publication in the Chemical Substance Index, and CTH is the code used to designate index entries intended for publication in the General Subject Index. The document analysts designate an index entry as a Chemical Substances entry (PAR) by keying a number in the NUM field. The numbers used in the NUM field must be unique for each chemical substance for a given document. These numbers are used only for CAS internal processing. The absence of a number in the NUM field causes the index entry in question to be generated as a General Subject entry (CTH).
5. A data field labeled CLASS. The function of this field will be described later.

Actually, the On-Line Phrase Input program contains a number of other data fields required by the CAS index production system in addition to those shown in the screen diagram in Figure 4. These data fields do not play a role in the phrase articulation process and are, therefore, not shown in the screen diagram or discussed here. However, the necessity of associating these various data fields with the appropriate headings played a major role in the design decision which resulted in the use of an individual screen for each manually flagged heading. Had it not been necessary to provide such linkages, the program could have been designed in a manner which would have allowed the flagging of all headings for a given phrase on a single screen.

Figure 4 illustrates the flagging required to designate SODIUM THIOCYANATE as a heading. The document analyst accomplishes this flagging by keying 3,4 in the HDG field. After keying all needed data on the FLAG HEADING screen, the document analyst hits the ENTER key to transmit the contents of the screen to the computer. Following this transmission, a new FLAG HEADING screen appears ready for the flagging of the next heading. In this manner, the

```
                    FLAG HEADING
                                          76678546X
effect of sodium thiocyanate on glycoproteins secretion in
 1   2   3        4       5       6           7     8

 tracheal mucus
    9     10

HDG: [6]
NUM:

                                     CLASS:
```

Figure 5. FLAG HEADING screen.

```
                    FLAG HEADING
                                          76678546X
effect of sodium thiocyanate on glycoproteins secretion in
 1   2   3        4       5       6           7     8

 tracheal mucus
    9     10

HDG: [10]
NUM:

                                     CLASS:
```

Figure 6. FLAG HEADING screen.

document analyst may flag as many headings in the phrase as he wishes as long as there is no overlap between flagged headings. The analyst indicates that he has flagged his last heading for the phrase by releasing (via the ENTER key) a FLAG HEADING screen with no keyed data.

Continuing with the example in Figure 4, if the document analyst wishes to flag a second heading in the phrase, he does so by keying the appropriate number(s) on a second FLAG HEADING screen. The flagging of a second heading, GLYCOPROTEINS, for this phrase is illustrated in Figure 5. Notice in Figure 5 that the numbers corresponding to the heading (SODIUM THIOCYANATE) flagged on the earlier screen are highlighted by the program. This highlighting allows the document analyst to maintain an awareness of words in the phrase which have been previously flagged as headings. The screen illustrated in Figure 5 is released to the computer and the program displays another FLAG HEADING screen. Figure 6 illustrates the flagging of a third heading, MUCUS. Notice in Figure 6 that the two previously flagged headings (SODIUM THIOCYANATE and GLYCOPROTEINS) are highlighted by the program.

Results Screen. The document analyst signals the program that he has completed the flagging of the last heading he wishes to flag for the phrase in question by releasing a FLAG HEADING screen with no keyed data. Following the release of the blank FLAG HEADING screen the program displays a RESULTS screen which shows an index entry generated by the program for each flagged heading. Figure 7 illustrates the RESULTS screen for the present example. The RESULTS screen displays the document identification number, the original phrase, and the list of algorithm-generated index entries. If there are more index entries than will physically fit on one RESULTS screen, subsequent "pages" of the RESULTS screens are generated by the program. The release of the last RESULTS screen causes the program to display a fresh ENTER PHRASE screen. At this point the document analyst may key an additional phrase for the same document.

```
                         RESULTS
                                               76678546X
effect of sodium thiocyanate on glycoproteins secretion in
  tracheal mucus


  Sodium thiocyanate
    glycoproteins secretion in tracheal mucus in relation to

  Glycoproteins
    secretion, in tracheal mucus, sodium thiocyanate effect on

  Mucus
    tracheal, glycoproteins secretion in, sodium thiocyanate
      effect on
```

Figure 7. RESULTS screen.

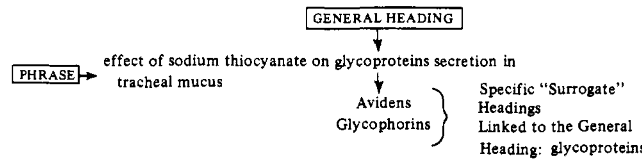

Figure 8. Generic-specific relationships.

```
                      SURROGATE
                                          76678546X
effect of sodium thiocyanate on glycoproteins secretion in
 1   2   3        4       5       6           7     8

 tracheal mucus
    9     10

CTH:  GLYCOPROTEINS
SURROGATE: [Avidins]
NUM: [1]

                         HEADING WORD NUMBERS: 6
```

Figure 9. SURROGATE screen.

Any number of phrases may be entered for one document.

Class and Surrogate Headings. Situations arise during the indexing of documents in which a "family" of index entries is generated, each with identical text modifications. The headings of specific members of such "families" of index entries often belong to a hierarchical class, the higher member of which is a generic term. Consider the example in Figure 8. Figure 8 contains the same phrase used in the earlier examples. Assume that we wish to flag the same three headings (SODIUM THIOCYANATE, GLYCOPROTEINS, and MUCUS). In this case, however, let us assume that the document being indexed contains information on two specific glycoproteins (AVIDINS and GLYCOPHORINS). The document analyst would, under these conditions, want to generate individual index entries for the two specific glycoproteins. These additional index entries would have identical text modifications. These text modifications will also be identical with the one generated for the class heading (GLYCOPROTEINS) in the phrase. The specific members of the hierarchical "families" are called surrogate headings. The generation of surrogate index entries is illustrated in Figures 9–11.

The document analyst designates a particular heading within a phrase to be a class heading by keying the code "C" in the CLASS field on the appropriate FLAG HEADING screen. In the present example, GLYCOPROTEINS would be designated as a class heading by keying a "C" in the class field of the screen illustrated in Figure 5. Having once designated a particular heading as a class heading, the program proceeds

```
SURROGATE
                                        7667854X
effect of sodium thiocyanate on glycoproteins secretion in
  1    2     3       4        5     6          7       8

tracheal mucus
  9       10

CTH: GLYCOPROTEINS
SURROGATE: [GLYCOPHORINS]
NUM:

                    HEADING WORD NUMBERS: 6
```

Figure 10. SURROGATE screen.

```
RESULTS
                                        7667854X
effect of sodium thiocyanate on glycoproteins secretion in
  tracheal mucus

 Avidins
   secretion, in tracheal mucus, sodium thiocyanate effect on

 Glycophorins
   secretion, in tracheal mucus, sodium thiocyanate effect on
```

Figure 11. Surrogate index entries.

to display special screens for the entry of surrogate headings to be linked to the class heading in question.

Figure 9 illustrates a SURROGATE screen displayed by the program as a result of the designation of GLYCOPROTEINS as a class heading. The SURROGATE screen contains:

1. The document identification number.
2. The numbered phrase with earlier flagged headings highlighted.
3. An indication of the class heading in the phrase to which the surrogate heading being entered will be linked (GLYCOPROTEINS).
4. A field for keying of the surrogate heading.
5. The NUM field used to distinguish Chemical Substance and General Subject index entries.

The only information keyed by the document analyst on the screen illustrated in Figure 9 is that shown enclosed in rectangles. All other information is displayed by the program. The document analyst may enter as many surrogate headings as he wishes for a particular class heading. Any or all of the headings flagged within a phrase may be designated as class headings and a "family" of surrogate heading entered for each.

Figure 10 illustrates the entry of a second surrogate heading for the class heading GLYCOPROTEINS.

In addition to the three index entries generated for the headings flagged within the phrase (shown in Figure 7), the program will generate an index entry for each of the two surrogate headings entered in Figures 9 and 10. These additional index entries are shown in Figure 11. Notice in Figure 11 that the text modifications for the two surrogate index entries are identical and that they are identical with the text modification generated for the class heading GLYCOPROTEINS (see Figure 7).

Inter-Screen Linkages. As the various screens used in the generation of index entries are processed, the OLPI program retains the data and linkages necessary to regenerate any of these screens upon receipt of appropriate commands from the

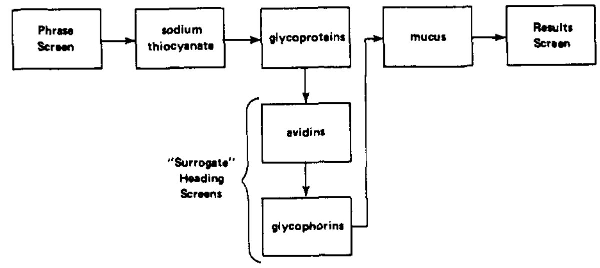

Figure 12. Inter-screen linkages.

document analyst. Figure 12 schematically represents the interscreen linkages set up by the program for the phrase used in the earlier illustrations. Figure 12 assumes the phrase flagged as indicated in previous examples including the class heading–surrogate grouping indicated earlier. Each rectangle in Figure 12 represents one of the screens illustrated earlier. Figure 12 represents the progression of screens from the original PHRASE screen to the final RESULTS screens.

Using the function keys mentioned earlier, the document analyst may, following the original pass through the set of screens represented in the Figure 12, retrace his steps through any or all of the earlier screens, regenerating each in the process. There are a number of reasons why document analysts may wish to return to earlier screens. He may wish to "unflag" a previously flagged heading. This can be accomplished by returning to the FLAG HEADING screen upon which the heading in question was flagged and then hitting a CLEAR key on the terminal keyboard. Another reason for regenerating earlier screens, which proved to be a powerful editing tool in the OLPI program, is the correcting of spelling or typographical errors. When a document analyst notices a misspelled word in a program-generated index entry, the analyst can regenerate the original PHRASE screen using an appropriate function key and then correct the misspelled word in the phrase. At this point, all index entries derived from the phrase instantly reflect the corrected version of the misspelled word. Using another function key, the document analyst is able to display the RESULTS screen at any point during the flagging of the phrase and thus view all of the program-generated index entries available at that time. He can subsequently flag additional headings or delete previously flagged headings.

Automatic Flagging of Headings. The automatic flagging option provides an alternative to the manual flagging of headings. The manual flagging option allows the document analyst to designate the headings within the phrase for which he wishes to have the program generate index entries. In the automatic flagging option, the program flags each character string within the phrase bounded by any one of the delimiters [a description of the articulation algorithm and a list of the word delimiters (mostly common prepositions) by which it operates is given by Cohen, Dayton, and Salvador[4]] recognized by the articulation algorithm or by the beginning or end of the phrase. The analysts may subsequently accept or reject any of the automatically flagged candidate headings.

The manual vs. automatic flagging decision for a particular phrase is made by the document analyst at the time the PHRASE screen (with a phrase keyed) is transmitted to the computer. It was mentioned earlier that the selection of the manual flagging option is signalled to the program when the document analyst hit the ENTER key following the keying of the phrase. If the document analyst wishes to auto-flag a phrase, he indicates this fact by hitting an appropriate function key following the keying of the phrase. The program then automatically flags a list of candidate headings as described above. It is important to note that the program generates interscreen linkages for automatically flagged headings. The

```
                    TRANSACTION
                                        76678546X
HDG:  SODIUM THIOCYANATE
TMD:  GLYCOPROTEINS SECRETION IN TRACHEAL MUCUS IN
        RELATION TO
ACT:      NUM: [1]

HDG:  GLYCOPROTEINS SECRETION
TMD:  IN TRACHEAL MUCUS, SODIUM THIOCYANATE EFFECT ON
ACT:  [X] NUM:

HDG:  TRACHEAL MUCUS
TMD:  GLYCOPROTEINS SECRETION IN, SODIUM THIOCYANATE
        EFFECT ON
ACT:  [X] NUM:
```

Figure 13. TRANSACTION screen.

Table I

ACT code	Function
D	Causes the candidate index entry to be deleted
X	Indicates to the program that it is necessary to go to the FLAG HEADING screen for the index entry in question (even though the heading has already been flagged automatically) for such purposes as manually altering the program-generated flags
C	Causes the heading in question to become a CLASS heading and signals the program to raise SURROGATE screens
"blank"	The absence of a code in the ACT field causes the index entry in question to be retained as a General Subject entry if the NUM field is blank, or as a Chemical Substance entry if the NUM field contains a number

linkages which the program would generate for the autoflagged index entries shown in Figure 13 are identical with those which the program would generate had the same three headings been manually flagged. The availability of the interscreen linkages for automatically flagged headings allows the document analysts all of the capabilities to redisplay any of the linked screens. In other words, FLAG HEADING screens may be displayed, at the option of the document analyst, for headings automatically flagged by the program.

Following the autoflagging of a phrase, the program displays a special screen called a TRANSACTION screen. The TRANSACTION screen which is generated by autoflagging the phrase used in the earlier examples is shown in Figure 13. The TRANSACTION screen displays the document identification number and the phrase. It also displays a candidate index entry for each of the autoflagged headings.

All of the fields shown in Figure 13 have been explained earlier with the exception of the action (ACT) field. The ACT field is unique to the TRANSACTION screen. It is used to communicate to the program a variety of signals relative to the disposition of individual candidate index entries. The document analyst may key any of the codes shown in Table I into the ACT field on the TRANSACTION SCREEN.

The list of candidate index headings generated by the automatic flagging technique does not always coincide exactly with the list of headings desired for a given document by the document analyst. Notice in Figure 13 that the second and third candidate index entries on the TRANSACTION screen are different from any of the manually flagged index entries shown on the RESULTS screen in Figure 7. Assuming that the index entries shown in Figure 7 are the desired entries, the codes shown (in the boxes) keyed in Figure 13 are those required to cause the program to display the appropriate screens to effect the necessary adjustments to the automatically assigned flags. The absence of a code keyed in the ACT field for SODIUM THIOCYANATE and the presence of a digit keyed in the NUM field would cause this heading to be retained as a chemical substance entry. The X's keyed in the ACT field of the second and third candidate index entries in Figure 13 would cause "preflagged" FLAG HEADING screens to be displayed for the two index entries in question. On these screens the document analysts can manually override the automatically assigned flagging digits. In the present example, the document analyst might select GLYCOPROTEINS and MUCUS as the two headings. Doing so would cause the words "secretion" and "tracheal" to be automatically placed at an appropriate position in their respective text modifications. In many cases it is not necessary to adjust any of the program-assigned flags, and the index entries can be accepted as they appear on the TRANSACTION screen.

Table II. Input-Output Statistics

No. of documents indexed	1450
No. of phrases input	2153
Average no. of phrases input per document	1.48
Total no. of index entries generated	7145
Average no. of index entries generated per document	4.92
Average no. of index entries generated per phrase	3.31

Vocabulary Control. The only vocabulary control which was part of the experimental phrase articulation program consisted of a stoplist of words which the program prevented from being autoflagged. Some effort has been made to develop vocabulary control techniques specifically designed to operate in conjunction with the On-Line Phrase Input program, but these techniques have not yet been incorporated into the program being reported on here. When fully developed and implemented, these techniques should, among other things, significantly enhance the autoflagging option. This will be accomplished by splitting multiple word autoflagged candidate index entries (such as the second and third entries in Figure 13) into one or more index entries with headings validated by authority files, thus reducing the number of required manual adjustments of automatically assigned flags.

Experimental Setting. Six document analysts from the CAS Biochemistry Department were trained in the operation of the On-Line Phrase Input program. The testing period for the on-line program was of five months duration. Document analysts indexed documents for their normal workflow during 2- to 3-h on-line indexing sessions at the terminal. In these sessions, document analysts indexed via the on-line phrase input method but carried out the other components of the document analysis task by standard methods. The document analysts graded the quality of program-generated index entries. The index entries resulting from the experiment were printed out for review but were not input to the CAS Index Production System.

RESULTS AND DISCUSSION

Training of Document Analysts. The training of the document analysts in the operation of the OLPI program involved:

1. Several brief instructional sessions of 1 to 2 h each (not at the terminal).
2. A small amount of basic printed reference material.
3. "Hands-on" practice sessions at the terminal.

Item 3 above was heavily emphasized and proved to be an effective training device both at the beginning of the experiment and later when program changes were implemented. All six of the document analysts involved in the OLPI experiment were able to utilize the on-line indexing method within the first few weeks of the beginning of the training sessions.

Overall Statistics. A total of 1450 documents representing 17 different CA sections were indexed during the On-Line Phrase Input experiment. No difficulties were encountered in applying the on-line indexing method in the various biochemical subject areas covered by the 17 CA sections involved in the testing.

Table III. Text Modification Quality Evaluation

Evaluation category	Grade	Percent
Perfect	1	76.6
Acceptable	2	19.9
Unacceptable	3	3.5

Table IV. Analysis of Spelling and Typographical Errors

No. of index entries examined	1325
No. of errors found	68
Error rate (on-line phrase input), %	5.1
Error rate (standard input method), %	12.0

Table II contains overall data on the number of phrases input and the number of index entries which resulted from the articulation of the phrases. The average number of index entries generated per input phrase is of particular interest since it indicates the keyboarding "leverage" which results from the generation of multiple index entries from the input of a single phrase. An average of 3.31 index entries was generated for each phrase keyed by the document analysts.

Study of Program-Generated Index Entries. a. Quality of Text Modifications. Table III contains the results of the evaluation of program-generated text modifications by the document analysts. A relatively high proportion (96.5%) of the program-generated index entries were judged by the document analysts to be sensible and intelligible. With experience, document analysts learned patterns for the construction of phrases which tend to produce a greater proportion of high-quality text modifications. Although the proportion of unacceptable text modifications generated by the articulation can be held to a fairly low level, it is apparent that an OLPI system will always produce some index entries in this category. Most Grade 2 and some Grade 3 text modifications can be upgraded to Grade 1 by a simple reordering of the clauses (segments of text modifications separated by commas) of which they are composed. It will be essential for any future production OLPI system to provide post-articulation editing capabilities which facilitate quick and easy on-line adjustments to program generated text modifications. Some work has been done on developing such post-articulation editing capabilities.

b. Analysis of Spelling and Typographical Errors. The results of a comparison of spelling and typographical errors occurring in index entries produced by the OLPI program and by the standard input methods are given in Table IV. A sample of 1325 program-generated index entries were examined for spelling and typographical errors. A total of 68 index entries (5.1%) were found to contain at least one such error. An earlier study of biochemical index entries generated by the standard dictation-keyboarding method indicated that 12% of standard-input entries required editing changes due to spelling and typographical errors. If a production scale OLPI operation were able to produce index entries with spelling and typographical errors on the order of 5%, the economics of index production would be favorably affected owing to the substantially reduced volume of required editing changes.

As mentioned earlier, the OLPI program allows spelling errors to be corrected in the phrase by going back to the PHRASE screen via a function key and then correcting the misspelled word. Such corrections are then immediately reflected in all index entries derived from the phrase in question. This feature of the OLPI program proved to be highly advantageous, particularly in cases in which large numbers of index entries were derived from the corrected phrase.

Probable reasons for the apparent reduction of spelling and typographical errors facilitated by OLPI are:

1. The absence of the recording-transcription interface facilitated by on-line input eliminates errors of interpretation occurring between the document analysts who dictate the index entries and the clerical staff who subsequently keyboard them.
2. The ability of the analysts to view the program-generated index almost immediately following the processing of the phrase, results in some fraction of the errors being noticed and corrected at the input stage.

Mitochondria
of brain, elongation of acyl-CoA by acetyl-CoA in, 76635674X
coenzyme Q10 semiquinone-succinate dehydrogenase complex of, of heart, ESR of, 76572210W
coproporphyrinogen oxidase of, in muscles of ascarid, 76606191E
cytochrome c reductase of, of Drosophila melanogaster mutants, heat effect on, 76569987E
DNS of kinetoplast and, structure of, 76623007K

Figure 14. Formatted program-generated index entries.

c. Study of Format and Appearance of Program-Generated Index Entries. The index entries from the OLPI experiment were sorted, formatted, and printed in a manner similar to the way index entries appear in the CAS volume indexes. The resulting listings facilitated the study of the appearance of program-generated text modificaton collected under specific headings. A small sample of program-generated index entries sorted and formatted in this manner is shown in Figure 14. The indexes resulting from the OLPI experiment appeared to resemble CA volume indexes very closely.

Systems Features and Human Factors. a. Function Keys. The function keys of the terminal were used to great advantage in the OLPI program. The ability to branch directly to various screens outside the normal forward processing path of the program was highly beneficial. This capability facilitated quick and easy changes in data keyed on earlier screens. The types of operations performed by the function keys are summarized below:

1. Forward and backward regeneration of screens
2. Scrolling through "multi-page" screens
3. Advancing directly to specific screens
4. Deleting previously entered flags
5. Carrying out specific operations (e.g., autoflagging)

In the original implementation of the experimental OLPI program, individual function keys were programmed to perform multiple operations. The particular operation carried out by a given function key depended on the point to which the analyst had progressed in the processing of the phrase at the time the function key was used. Multi-operation function keys proved to be a definite problem. The difficulties came about when an analyst hit a function key expecting a particular operation to be performed, but found that some completely different operation had taken place. Often this resulted in the analyst finding a particular screen displayed on the CRT and not understanding what had caused the screen to appear. This kind of experience was quite frustrating to the document analysts, who expressed the feeling of being "lost" in the flow of screens.

Two steps were taken to alleviate the function key problem. With two exceptions the function key operations were modified such that each key performed a single operation only. In the two cases in which function keys were allowed to retain dual operations, the functions were analogous to the degree that they caused no confusion to the document analysts. As presently programmed, individual function keys are applicable only at particular points in the processing of a phrase and only when certain screens are being displayed on the CRT. Coincident with the modification of the function key operations, a group of interactively applied edits were added to the OLPI program which caused error messages to be displayed any time a particular function key was invoked at an inappropriate point during the processing of the phrase. The

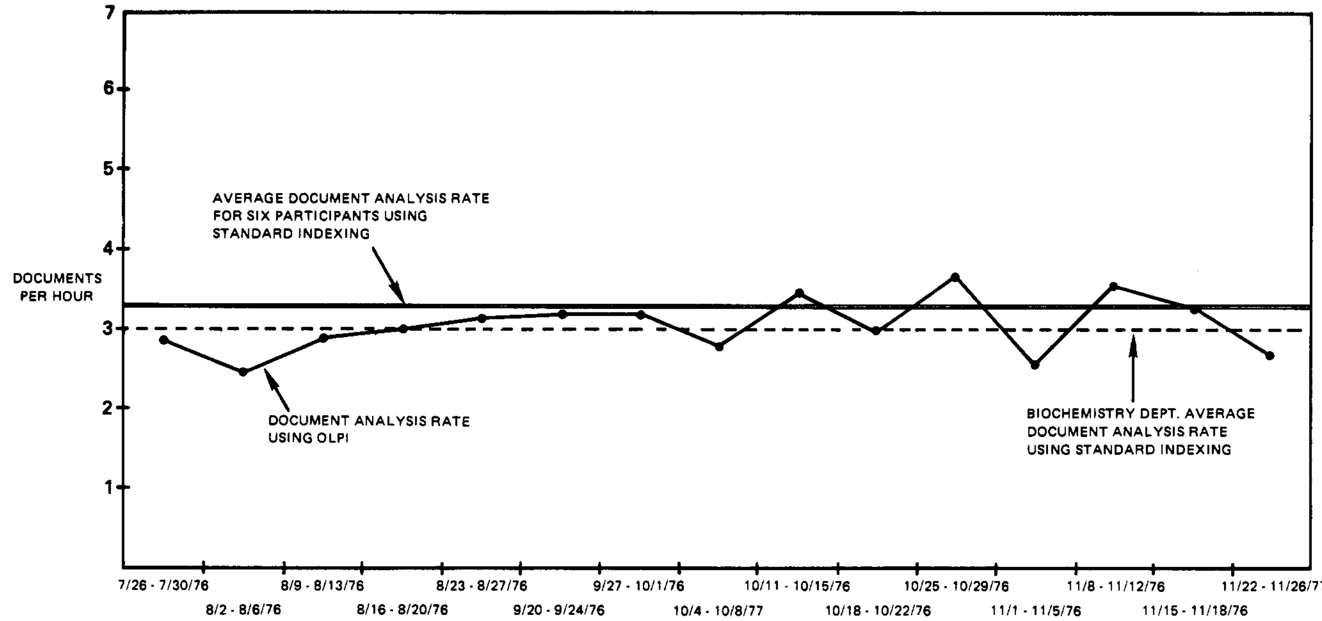

Figure 15. Production rates for document analysis including indexing by on-line phrase input and by standard methods.

combination of the modified function key operations and the interactively applied edits eliminated the "getting lost" problem described above. These changes represented a major operational improvement of the OLPI program.

b. Screen Layout. The screens in the OLPI program were designed such that the most frequently utilized fields were placed generally toward the top of the screens. Other fields were placed generally lower on the screens in decreasing order of their frequency of use. This arrangement of fields tended to minimize the amount of tabbing past unused fields on particular screens.

c. Factors Relating to the Terminal. One significant problem with the IBM 3277 keyboard which became apparent during the experiment resulted from the close physical proximity of the SHIFT and ENTER keys. The ENTER key is the most basic function key and is used for most of the normal forward processing. The ENTER key was sometimes accidently struck when the analysts had intended to hit the SHIFT key. This caused the program to display screens which the analysts did not expect to see at the time they appeared. Again, the net effect was a sense of being temporarily "lost" in the flow of screens. Fortunately, the document analysts were usually able to recover by retracing their steps through the use of appropriate function keys.

The analysts found the terminal display (sharpness of characters, spacing of lines, etc.) to be quite good. The terminal emits a high-pitched tone which three of the six participants in the experiment found quite disturbing. The terminal noise did not present any problem to the other three document analysts.

Other characteristics of the terminal which proved to be particularly beneficial in the operation of the OLPI program are:

1. Capability of having data appear on the screens with different light intensities. This feature was used to highlight previously flagged headings on FLAG HEADING screens.
2. Automatic skipping of the cursor past protected fields on screens.

d. Fatigue Problems. The experiences of the six analysts relative to fatigue resulting from the OLPI operation were mixed. Three of the six document analysts did not find fatigue to be a problem with OLPI. Two of the document analysts thought that OLPI fatigue was greater than that experienced in another on-line application at CAS. One factor which tended to lessen fatigue was the intermittent terminal use pattern of the OLPI operation. Only about 25% of the analysts' time was spent doing on-line indexing. The other 75% of their time was spent performing the other components of the document analysis task. One analyst felt that the intermittent terminal use pattern of the OLPI operation was considerably less fatiguing than the steady use pattern of other on-line programs with which he had experience.

e. Response Time. The response time of the experimental OLPI program was usually well within the acceptable range[7] for such an activity (under 5 s and frequently almost instantaneous). Occasionally, response time reached 5 to 10 s, and this range did begin to be a concern to the document analysts. Generally, response time was not a problem during the experiment.

f. Edits and Consistency Checks. The experimental OLPI program proved particularly strong in the area of error detection, prevention, and correction. Contributing factors were the following.

1. Interactively applied edits applicable to individual fields on screens under specific conditions prevented program-detectable erroneous or inconsistent data from ever being processed. These edits played a major role in the elimination of the errors and inconsistencies of the earlier batch dictation phrase input work (Cohen, Dayton, and Salvador[4]).
2. The absence of the document analyst-keyboarder interface eliminated errors of interpretation occurring at that interface.
3. Display of index entries on the CRT facilitated detection of errors, and resulted in relatively "clean" index entries at the input stage, diminishing the volume of subsequent editing changes.
4. The OLPI program facilitated a substantial "leverage effect" in the correction of misspellings and typographical errors (i.e., corrections made once in the phrase were immediately reflected in all index entries derived from the phrase).

g. Production Rates. An average of about three documents per hour were processed by the document analysts participating in the OLPI experiment. About 5 min per document was required for the on-line indexing of the documents while the other components of the document analysis task required about 15 min per document.

Production rates for the six participants in the experiment are give in Figure 15. This graph compares document analysis rates for indexing done via OLPI to rates in which the indexing portion of the document analysis task was done by the standard indexing method. Comparisons are given in Figure 15 for both the six participants in the experiment and the CAS Biochemistry Department as a whole. The production rates measured during the OLPI experiment were not significantly different from average production rates of the standard method.

During the course of the experiment frequent and substantial changes were made in the OLPI program. These changes necessitated retraining the document analysts. Longer experience with a stable program would be expected to result in some increase in production rates. Another factor which tended to reduce production rates during the experiment was the task of grading the quality of the text modification produced by the on-line production. This task represented additional intellectual effort required of the document analysts during the experiment, which would not be required in a production-scale on-line indexing operation. Because there was no practical way of separating the time required for the grading task from the remainder of the indexing effort, this time component is reflected in the data derived from the experiment.

Input Phrases and Machine Search Files. Utilization of input phrases in search files appears to be a potentially valuable by-product of the phrase input process. The input phrases are relatively nonredundant compared to the index entries of which CASIA (Chemical Abstracts Subject Index Alert) is now composed, and thus hold the potential of dramatically reducing the volume of data contained in the search files with no loss of searchable information.

CONCLUSIONS

1. The experimental OLPI program in its present form is highly workable from an operational viewpoint. Optimization of human factors contributed substantially to the general workability of the program. The fine details of the human-machine interface deserve careful attention in the design of any possible future production scale OLPI program.

2. Document analysts were able to learn the operation of the OLPI program after a small amount of instruction and several weeks of "hands on" practice sessions. No serious problems were encountered by the document analysts in adapting to the on-line input environment. The variety of subject matter contained in the 17 CA Biochemistry sections covered by the experiment presented no difficulties to the document analysts in applying the interactive indexing method.

3. A high proportion of index entries of acceptable quality was generated by the OLPI program. A production-scale OLPI system will require post-articulation editing to facilitate the upgrading of program-generated index entries judged unacceptable by the document analysts. Program features which allow the document analysts to interactively alter the order of the clauses of which program-generated text modifications are composed will be essential in an OLPI production system.

4. Production rates for the total document analysis task were about the same using the experimental OLPI programs as production rates involving standard indexing. But during the course of the exeriment, document analysts were frequently required to adjust to changes in the program. Longer experience with a stable program would be expected to result in some increase in production rates.

5. The interactive edits of the experimental OLPI program, which prevented document analysts from proceeding until the error conditions were corrected, were a powerful device in obtaining relatively "clean" index entries at an early stage in the total index production process. These edits were also an important factor in the elimination of the coding inconsistencies which were a major problem in the earlier batch dictation phrase input work. This type of interactive edit should be exploited in the design of any future production OLPI system to the extent that input errors and inconsistencies are program-detectable.

6. OLPI appears to afford a substantial reduction of typographical and spelling errors compared to the present dictation–transcription input method. The on-line indexing method may, therefore, hold the potential for reducing the volume of editing changes required.

7. Function keys are of great value in the OLPI operation. The analysts' ability to transfer their attention directly to specific screens outside the normal forward flow is an important capability which facilitates correction of data keyed on earlier screens. Programmed function keys are a highly desirable method of providing these capabilities.

8. An index formed from the program-generated index entries was found to closely resemble the appearance and style of the CAS Volume Indexes.

LITERATURE CITED

(1) J. E. Armitage and M. F. Lynch, "Articulation in the Generation of Subject Indexes by Computer", *J. Chem. Doc.*, **7**, 170–8 (1967).
(2) J. E. Armitage and M. F. Lynch, "Some Structural Characteristics of Articulated Subject Indexes", *Inf. Storage Retr.*, **4**, 101–11 (1968).
(3) R. Salvador, "Automatic Abstracting and Indexing", Masters Thesis, The Ohio State University, Columbus, Ohio, 1969.
(4) S. M. Cohen, D. L. Dayton, and R. Salvador, "Experimental Algorithmic Generation of Articulated Index Entries from Natural Language Phrases at Chemical Abstracts Service", *J. Chem. Inf. Comput. Sci.*, **16**, 93–99 (1976).
(5) B. Hall, "A Computer-Generated Index Technique", *Indexer*, **8**, 130–138 (1973).
(6) Paper entitled, "Cost Effectiveness of an On-Line Editing System at Chemical Abstracts Service", K. H. Baser, W. G. Cox, F. K. Salinshick, and R. L. Wigington, presented at the American Society for Information Science, 6th Mid-Year Meeting, 19 May 1977, Syracuse University, Syracuse, N.Y.
(7) J. Martin, "Design of Man-Computer Dialogues", Prentice-Hall, Englewood Cliffs, N.J., 1973, Chapter 18, pp 321–332.