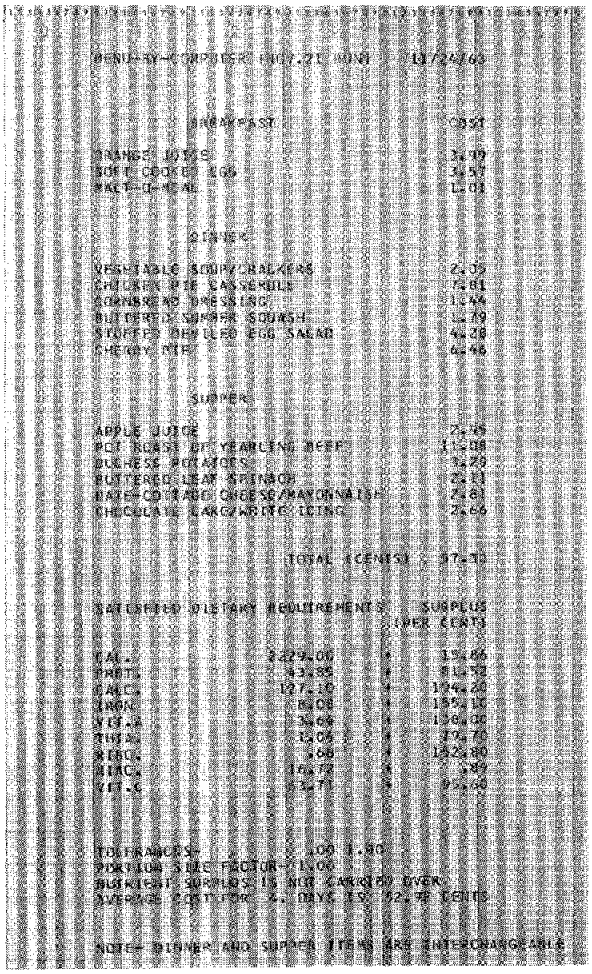

FIG. 3

the 1440 system is under consideration. The program can be easily adjusted to solve a large variety of menu problems with different sets of objectives.

Acknowledgments. The author acknowledges the cooperation of Tulane Bio-Medical Computing System and Touro Infirmary, New Orleans, on the project with special appreciation for the support of Dr. James W. Sweeney, Co-Principal Investigator.

RECEIVED DECEMBER, 1963; REVISED JANUARY, 1964.

REFERENCES

1. STIGLER, G. J. The cost of subsistence. *J. Farm Economics 25* (1945), 303–314.
2. SMITH, V. E. Linear programming models for the determination of palatable human diets. *J. Farm Economics 41* (1961), 272–283.
3. WATT, B. K., MERRILL, A. L., ORR, M. L., ET AL. Composition of foods—raw, processed and prepared. U. S. Department of Agriculture Handbook No. 8, 1950.
4. PERYAM, D. R., POLEMIS, B. W., KAMEN, J. M., EINDHOVEN, J. AND PILGRIM, F. J. Food preferences of men in the U. S. Armed Forces. Dept. of the Army, Quartermaster Research and Engineering Command, Quartermaster Food and Container Institute for the Armed Forces, Jan. 1960.
5. Recommended dietary allowance. Natl. Research Council, Food and Nutrition Bd., Natl. Research Council Publ. 589, Rev. 1958.
6. DANTZIG, GEORGE B. *Linear Programming and Extensions.* Princeton U. Press, Princeton, 1963.

Information Retrieval

H. R. KOLLER, Editor

Problems in Automatic Abstracting

H. P. EDMUNDSON
The Bunker-Ramo Corporation, Canoga Park, California

A variety of problems concerning the design and operation of an automatic abstracting system are discussed. The purpose is to present a general view of several major problem areas. No attempt is made to discuss details or to indicate preferences among alternative solutions.

1. Introduction

Since automatic abstracting is in its infancy it is felt that a paper covering the subject as a whole is apt to be more helpful than one which pleads for a single course of research. In many ways the present situation in automatic abstracting in the United States is analogous to the early days of automatic translation. For example, only two or three research teams, totaling 12 people, are presently working on the problem of automatic abstracting, while a dozen teams with a total staff of some 100 researchers are now studying automatic translation. Moreover, various United States government agencies have invested several millions of dollars in automatic translation since 1953, while only several hundred thousand dollars have been made available for research on automatic abstracting since 1958.

In this exposition the problems of automatic abstracting are grouped into the following major classes: (1) conceptual problems, (2) input problems, (3) computer problems, (4) output problems, and (5) evaluation problems.

Present systems of automatic abstracting are capable of producing nothing more than extracts of documents, i.e. a selection of certain sentences of a document. This is not to say, however, that future automatic abstracting systems cannot be conceived in which the computer generates its own sentences by means of a suitable generative grammar program. Theoretically there is no linguistic or mechanical reason why such a system could not be designed and operated. The total system would then consist of a program which operates on the original document so as to produce an extract which in turn is fed into the generative grammar portion that then generates its own sentences using certain of the original sentences as grist This system is depicted in Figure 1. Such a system, however, is apt to be costly both in time and money.

Since the creation of a suitable generative grammar

This research was supported in part by the United States Air Force with funds from Contract No. AF 30(602)-2223, monitored by the Rome Air Development Ctr., Griffiss Air Force Base, N.Y.

program lags somewhat behind that of abstracting programs, attention here is confined to automatic abstracting systems that involve only extracting.

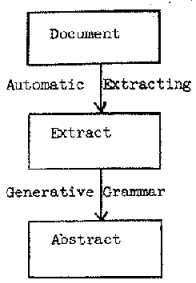

FIG. 1

2. Conceptual Problems

DEFINITION OF AN ABSTRACT. Assume that an extract of a document (i.e. a selection of certain sentences of the document) can serve as an abstract. In defining such an abstract of a document we must specify the following three aspects: content, form and length. The problem of content in an automatic abstract is that of selecting or rejecting sentences of the original document so as to form an acceptable extract or abstract. The problem of form is that of deciding how the selected sentences are presented to the reader in relation to the formatting of the title, authors, headings and subheadings, graphics, footnotes and references. The problem of length is that of deciding how many words or sentences will constitute the final output according to fixed rules, variable rules and thresholds of compactness.

An interesting way to view the length of an abstract is to compare it with its sister categories—document, title and index term. If these four categories are ranked in increasing length, in terms of either words or bits of information, the order becomes: index term, title, abstract, document. Moreover, considering the lengths of these four categories to within an order of magnitude, one observes the geometric progression 1, 10, 10^2, 10^3. In other words, an abstract is approximately 10 times the length of the title and approximately 1/10 the length of the document. Seen as a whole this geometric progression represents the increasing degree of condensation of information-ranging from the document, through abstract and title, to the index term.

It is currently believed that the notion of the abstract of a document is simple and generally understood, i.e. that to every document there corresponds one abstract. To put it mathematically, the abstract A is a function of the document D, i.e. $A = f(D)$. Moreover, since an abstract is here an extract, A is a subset of D, i.e. $A \subseteq D$.

However, on closer examination it may be seen that a document can and does have many abstracts which differ from one another not only in content, length and format, but also in their intended use. Hence, the act of abstracting is goal-oriented. With the realization that it is misleading to conceive of *the* abstract, we must now speak of *an* abstract of a document. Thus, an abstract is a function of the two quantities, the document D and the use U, i.e. $A = f(D, U)$.

Despite the fact that the preceding observation is simple and intuitively acceptable, its consequences are neither of these. In fact, it provides the foundation for a solution to the problem of defining an automatic abstract. Because of various alternative uses, it is necessary to define "abstract content" explicitly in terms that are use-oriented. This definition must be expressed by machine criteria. To do this requires detailed specification far beyond what might initially have been expected. Thus, we seek to eliminate arguments over what is an abstract by replacing useless generalities with specific operational criteria.

DEFINITION OF A GOOD ABSTRACT. This problem is closely related to the section devoted to evaluation of the quality of abstracts. It involves questions of the existence of a completely general definition of an abstract versus that of many specific definitions.

This leads to the concept of a tailor-made abstract, in the sense that an individual will be able to specify in future automatic systems more accurately what he wants in an abstract. Moreover, this feature distinguishes automatic abstracting from automatic translation. It is widely accepted that, aside from minor stylistic variations, there is only one translation of a document. On the other hand it has been shown that a document can have several different abstracts. This difference is fundamental to the problem of evaluating the quality of automatic abstracts, and supports the general feeling that the problem of evaluating translations is considerably easier than that of evaluating automatic abstracts.

RESEARCH METHODS. Problems here concern the set of techniques that are used to guide the research effort in automatic abstracting. For example, such problems are encountered as how to improve intermediate products by iterative techniques, how to specify or describe linguistic and statistical clues of textual behavior, and what general principles are to be followed as guide lines. Among the several principles, we stress one that seems dominant.

Principle 1. Employ a method that detects and uses all abstracting clues (e.g. of meaning, significance, organization, etc.) provided by the author, the editor and the printer.

This principle focuses on capturing automatically as many clues as possible that are, either consciously or unconsciously, provided by the creators of the document. For example, the skilled author selects an appropriate title, organizes his thoughts in distinct sections with appropriate subtitles, condenses much information in the captions of graphs and tables, and uses footnotes and references in revealing ways.

It is instructive to regard the problem of automatic abstracting in the light of several other principles.

Principle 2. Employ mechanizable criteria of selection, i.e. a system of rewards for desired sentences.

Principle 3. Employ mechanizable criteria of rejection, i.e. a system of penalties for undesired sentences.

Principle 4. Employ a system of parameters that can be adjusted in order to permit tailor-made abstracts.

Principle 5. Employ a system which is a function of several distinct factors, such as statistical, semantic, syntactic, locational, etc.

3. Input Problems

THE CORPUS. Documents taken from a particular corpus or body of text may have important similarities among one another and important dissimilarities with documents taken from a different corpus. Thus, one of the first problems in conducting research in automatic abstracting is that of choosing an appropriate corpus. For example, problems arise due to the subject matter (e.g. sociology vs. mathematics), the publishing medium (e.g. newspaper vs. text books), editors' rules regarding acceptability for publication (e.g. research papers vs. expository works), and the author's style and compactness of presentation.

PRE-EDITING. The above remarks place difficulties in the path of the pre-editing step since at the present time one must resort to keypunching the original document. Moreover, even when print readers are available they may not be equal to the task. Hence, the text must be manually pre-edited according to a set of pre-editing instructions. The creation of these instructions is not trivial because it is precisely at this step that a choice may be made to retain or ignore those critical format clues which, once lost, can never be restored by any programming tricks. The pre-editing instructions must cover problems of formatting, graphics, special symbols, special alphabets, footnotes and references.

KEYPUNCHING. Despite the fact that keypunch operators quickly adapt to new problems, it is necessary to prepare a set of keypunch instructions. These instructions are based upon the pre-edit instructions and are subject to the boundary conditions imposed by available input and output hardware. They must contain rules of sufficient generality to cover a wide variety of textual situations and should also be supported by appropriate examples. The purpose of these keypunch instructions is to minimize decision making by the keypunch operator.

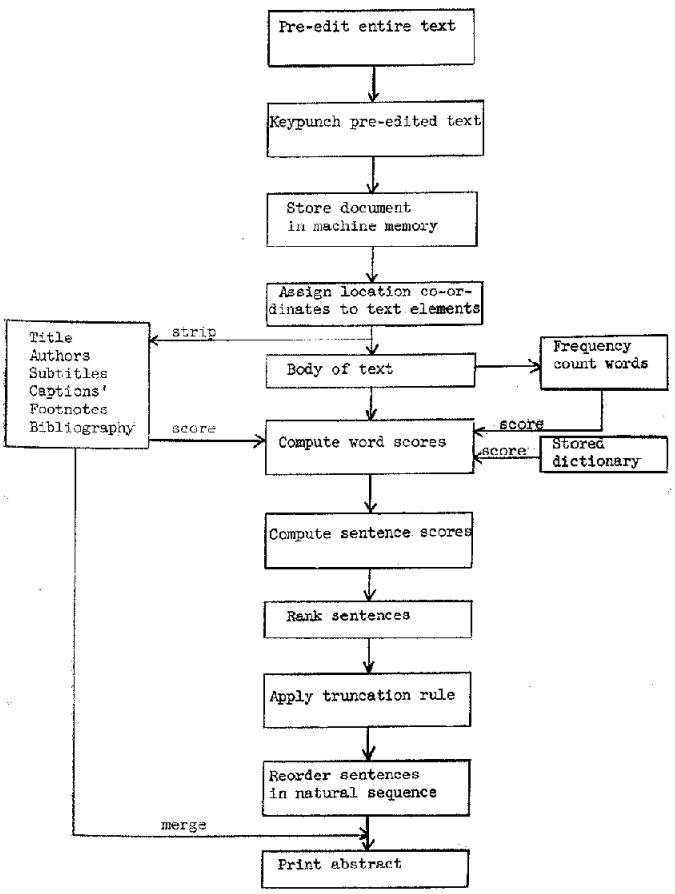

FIG. 2. A possible automatic abstracting system

4. Computer Problems

SYSTEM ASPECTS. By "system aspects" we refer to the functional specification of each of the steps in the automatic abstracting system. In general these steps are: pre-editing the textual input, assigning proper sequence numbers to successive elements of the text, weighting the textual factors according to some scheme, scoring the text sentences by combining these weights, ranking the sentences in decreasing magnitude, truncating this decreasing sequence at some threshold and outputing the set of sentences (Fig. 2).

In accordance with Principle 2, instead of using the loaded words "topic" or "significant," as has often been done, to refer to the sentences chosen out of the original article for an abstract, a neutral name might be chosen, such as "A-sentences" for those that are considered acceptable. The use of this notation for the chosen sentences lends itself very nicely to further operations. For instance, think of the set A_i as being those sentences chosen by factor S_i. Similarly, having another group of sentences chosen by factor S_j, this second set of sentences is denoted by A_j. It would then be possible to consider sentences selected by factor (S_i or S_j) or by factor (S_i and S_j). An extension of this notion would be to consider the set of sentences A_S where vector S is a vector of selection factors. If T is another vector of different selection factors, all the sentences chosen by S and by T could then be compared.

Another use of the A-notation would be to denote the body of sentences chosen by different stages in the selection process, assuming that it is desired to break the selection process up into stages. If this were done, then A_0 could be considered to be the entire original article, A_1 to be those sentences chosen by the first stage of the selection process, A_2 to be those sentences chosen by the second stage, and so on. Thus, A_1 is the result of applying the transformation T_1 to A_0, i.e. $A_1 = T_1(A_0)$, A_2 is the result of applying transformation T_2 to the set of sentences A_1, i.e. $A_2 = T_2(A_1)$, etc. Similar uses of this notion will probably suggest themselves.

It is possible to apply various kinds of selection criteria. It might be desired, for instance, to select by the first stage selection process (producing the set of A_1 of sentences) all the sentences which were not rejected by some particular rejection criterion. (Note the use of rejection criteria here as opposed to the acceptance criteria customarily used.) Another candidate for first-stage selection might be the use of only nonstatistical criteria for the first stage of selection, followed by only statistical criteria for the second stage, or the reversal of the order of these two steps.

Certain sentences might be chosen as being among those which must be included in the abstract. In fact, the title of the article may be one of these. Such sentences could be selected and then set aside in an untouchable body of sentences so that they could not be rejected by any further processing. The selection process could consist of repetitions of the same kind of transformations on the body of sentences, and the process would end when $A_{n+1} = A_n$; that is, when the sequence of reductions converged to a minimal set of sentences. It would be necessary, of course, for such a set of reduction processes to insure that not all sentences were eliminated!

In accordance with Principles 3 and 4, one can view, in a statistical framework, the problem of selection of sentences for an abstract as the problem of selecting the right answers versus wrong answers. By "right answers" we mean those sentences which one would want in an abstract, and by "wrong answers" those which one would not want. Clearly, in any article there are sentences which should be included in every abstract and there are sentences which should not be included in any abstract. The fact that there might be a large number of indeterminate answers is not the issue at the moment. The problem of selecting sentences for an abstract is that of holding the number of false answers to a minimum while selecting as many as possible of the right answers. In other words, this is the familiar statistical problem of trying to place the level of acceptance at such a point that the desired number of right answers is chosen and, at the same time, as many as possible of the wrong answers are rejected.

In accordance with Principle 5, one way of selecting sentences for an abstract is by means of various factors, and combining them to form a single factor T. Suppose the S_i's denote semantic factors, syntactic factors, locational factors, editorial factors and so on. To each S_i associate a weight w_i, and form the linear combinations $T = \sum_i w_i S_i$ of the products $w_i S_i$. The parameters w_i then can be adjusted to reflect the relative importance of the factors S_i.

An extension of this idea is that different sets of weights, i.e. different vectors w of weights w_i could be formed, with a different column vector of weights for different journals. Abstracting an article from a given journal then would have as one of its steps the selection of the proper set of weights w for use in an otherwise general program. A possibility deriving from this approach is that row averages could be taken of the components of all these column vectors, and the vector of row averages could be used as a reasonable weighting scheme for abstracting an unfamiliar journal.

PROGRAMMING. Problems here concern the nature of individual routines and subroutines. For example, it is useful to separate the total system into three major operating programs: edit program, dictionary program, and abstracting program. In addition to these operating programs various research programs must be written. Based upon the theoretical model or structure underlying the abstracting system, decisions must be made as to the best method of using a mixture of computing routines and table-lookup routines. The abstracting system should provide for the readjustment or modification of the numerous parameters that are incorporated in the programs or that are stored in the tables. This allows discoveries made during periods of research to be easily transformed into improvements in the operating programs.

TABLES. The success of an automatic abstracting system depends materially upon two different aspects. The first aspect concerns the general system or method of abstracting as given by the sequence of programming operations. The second concerns the specific entries of the several stored tables. An example of a stored table is a glossary or dictionary of several thousand words that act either as cue words that signal the importance of a sentence, or as stigma words that signal the unimportance of a sentence for purposes of abstracting. Such a table may include, in addition to the word, a code indicating its grammatical or semantic function, its importance weight, etc. Another kind of table may be set aside to retain the title, author, section headings, footnotes and references awaiting use during the final output step of the program. A third possibility is the inclusion of a table of synonyms and antonyms which will handle some semantic problems via the thesaurus method. In any case the programmer is presented with the sizable problem of juggling sections of the internal memory in order to accommodate the input text, the program and the tables.

5. Output Problems

HARDWARE. As in the case of input, we are confronted with problems imposed by output hardware. Despite the fact that high speed printers are available, the most serious difficulty is that of an over-restricted number of type fonts. This forces a replacement of strings of unusual symbols (e.g. mathematical and chemical) by the few conventional symbols available at the output printer. Moreover, important segments of textual symbols are also forced to be replaced by only one or two such conventional output symbols.

A second problem here is that of composing. Present output hardware provides little leeway in the composition of the textual output.

FORMAT. The format of the classical, human-prepared abstract comprises title, author and a paragraph of connected text. However, since present automatic abstracts are in fact nothing more than automatic extracts, it is desirable to correct the generally disjointed sequence of selected sentences by other devices. This problem can be partially solved by capturing in an automatic abstract those informative features of structure found in section headings and subheadings, together with footnotes and references.

DISSEMINATION. Despite the fact that the problem of dissemination of automatic abstracts has received little attention in the literature, it nevertheless will play an important part in the general acceptability and utility of automatic abstracts. Both theoretical and practical studies must be made to ascertain how the requestor communicates with the abstracting system, how the system collates

similar requests, and how the system produces and disseminates multiple copies of the abstracts through a suitable medium and communication channel.

6. Evaluation Problems

ACCEPTABILITY. The first problem of evaluation concerns the acceptability or utility of the final product. This customarily requires that some qualitative or quantitative comparison be made between an automatic abstract and an "ideal" human abstract. However, it is of interest to note that repeated experiments conducted among human abstractors have revealed that the linear coefficient of correlation among humans varies from .2 to .4, even when they have operated under moderately well-defined abstracting rules. This disappointing result, although not totally unexpected, is due in part to the fact that the correlation coefficient is not the best measure. For example, if two individuals happen to select different, but cointensional sentences, then the correlation coefficient will naturally be low. The problem of what sentences of a document are cointensional is solvable only by further semantic research which, unfortunately, has yet to be done. The generally poor interhuman agreement tends to force us in the direction of arbitrarily, but uniformly, defining what an abstract is and then mechanizing these properties.

COST. The second problem of evaluation is that of system cost in dollars and in time. At present, insufficient concrete data have been collected to permit reliable estimates of cost per document and estimates of bounds on the error for an operating system. However, such information does exist for research systems that do not claim operational perfection.

7. Remarks

In spite of the problems highlighted above it is felt that automatic abstracts can be defined, programmed, and produced in an operational system so as to complete with present human abstracting. The basis for this optimism is the fact that several automatic abstracting systems are presently producing abstracts, regardless of how unsophisticated they may be. That the future automatic abstracts will be different both in content and appearance from classical ones seems clear. However, it is not expected that users will be materially inconvenienced by having to adapt to a new format. Further research needs to be performed in this area of linguistic data processing, but the true nature of this problem is being seen clearly for the first time.

RECEIVED JULY, 1963; REVISED SEPTEMBER, 1963.

REFERENCES

1. LUHN, H. P. The automatic creation of literature abstracts. *IBM J. Res. Develop. 2*, 2 (April 1958).
2. Final report on the study of automatic abstracting. C107-1U12, Thompson Ramo Wooldridge Inc., Canoga Park, Calif., Sept. 1961.
3. EDMUNDSON, H. P., AND WYLLYS, R. E. Automatic abstracting and indexing—survey and recommendations. *Comm. ACM 4*, 5 (1961) 226-234.

LETTERS—*Continued from page 203*

language documents, programs, texts of telegraphic messages, etc.) to perform a variety of functions (verification of indexing, vocabulary, automatic translation, label checking, etc.).

Despite Mr. Radford's assertion that his suggestions meet "coldly scientific requirements," there seems no doubt that a rule (3(c)) which contains the phrase "pronounciation is made more obvious" is an invitation to inconsistency. Nor is it clear what an "accepted" combination (rule 2) might be. Unfortunately for those of us who have poor intuitions about accepted combinations, the example given for rule 2 occurred at the end of the line in the text of Mr. Radford's note and was therefore hyphenated! How did it appear in the manuscript?

The point is simply that under our present standards for hyphenation and their use by humans, inconsistencies do occur. In any production operation on a computer these generate either failures in the matching operation or require relatively complicated programming tricks to bring together the alternative spellings.

The difficulty with hyphens is that there is no single way that they can be used with consistency. Nor, for that matter, can one state unequivocal rules for the use of intervening spaces. Miss Grems' suggestion to combine terms has at least the virtue of consistency. The fact that it saves a few characters here and there is incidental.

T. R. SAVAGE
Documentation, Inc.
4833 Rugby Rd.
Bethesda 14, Md.

Empirical Bounds for Bessel Functions

Dear Editor:

This note is concerned with the article published in *Communications 1*, 5 (May, 1958), entitled "Note on Empirical Bounds for Generating Bessel Function" by James B. Randels and Roy F. Reeves.

For $J_n(X) = KJ_n^*(X)$, read $J_n(X) = \dfrac{J_n^*(X)}{K}$;

for $K = J_0^*(X) + 2\sum_{n=1}^{\hat{n}} J_{2n}^*(X)$,

read $K = J_0^*(X) + 2\sum_{n=1}^{\hat{n}/2} J_{2n}^*(X)$;

for $Y_0(X) = \dfrac{2}{\pi}\left[J_0(X)\left(\gamma + \ln\dfrac{x}{2}\right) - 2\sum_{n=1}^{\hat{n}}(-1)^n J_{2n}(X)\right]$,

read $Y_0(X) = \dfrac{2}{\pi}\left[J_0(X)\left(\gamma + \ln\dfrac{x}{2}\right) - 2\sum_{n=1}^{\hat{n}/2}\dfrac{(-1)^n J_{2n}(X)}{n}\right]$.

With these revisions Bruce Lemm and I have developed a double precision (IBM 7090) code that *supports* observation 2 in the conclusion section; i.e., for all values of $J_n(X)$ and $Y_n(X)$ where $0 \leqq n \leqq 9$, and $0.1 \leqq X \leqq 25$ the *generated values* agreed with those in the British Association *Table of Bessel Functions* to a *maximum* error of 1 in the sixth *significant* digit whenever the solution was greater than 0.1. Moreover, using the Harvard Tables of $J_n(X)$ this conclusion is valid for $0.1 \leqq X \leqq 100$. Whenever the solution is less than 0.1, the answer suffers a greater loss in significant figures.

R. L. PEXTON
University of California
Lawrence Radiation Laboratory
Livermore, California