American Journal of Computational Linguistics *Microfiche 4*

A Few Steps Towards

Computer Lexicometry

Nicholas V. Findler and Heino Viil

Department of Computer Science
State University of New York
Buffalo

A FEW STEPS TOWARDS COMPUTER LEXICOMETRY

Nicholas V. Findler and Heino Viil
Department of Computer Science
State University of New York at Buffalo

ABSTRACT

We describe a branch of dictionary science, and recommend the term lexicometry for it, that deals with the mathematical and statistical aspects of dictionaries. It is related to both lexicography and lexicology, the former denoting the description of lexical material and the latter its analysis and study.

Many problems in computational linguistics require the use of a stored dictionary easily accessible to a computer program. In the course of an investigation, such a dictionary may have to be expanded, reduced, rearranged, or modified in various ways. Also several nonlinguistic disciplines using the computer, such as psychology, biology, medicine, and sociology, often need a large data base in the form of a dictionary. The relevant structural properties of a dictionary, however, have not yet been sufficiently and systematically investigated. Research in this area is needed in order to optimize the construction of stored dictionaries and to manipulate them in efficient ways.

---

[1] A considerably extended version of this paper was submitted to the State University of New York in Buffalo in partial satisfaction of the requirements for the degree of Master of Science of Heino Viil. The project represents the continuation of an earlier work by Nicholas V. Findler. Many ideas and all the programming effort is due to Heino Viil. The write-up is a joint effort. The work reported here was supported by National Science Foundation Grant GJ-658.

First, we review critically the problems of meaning and its representation, the questions relating to lexical definitions. to polysemy, homonymy, semantic depletion. synonymy, and lexicography and lexicology in general. We also discuss the concept of lexical valence and elaborate a novel idea, coverage, which is of both theoretical and practical importance. In this context, relationships are established among three variables: the size of the covered set, the size of the covering set, and the maximum definition length. Both, the size of the covering set and the maximum definition length should be small for economic considerations. But decreasing one will increase the other. It is therefore important to establish these relationships empirically. The knowledge so gained will constitute a basis for optimizing the structure of a dictionary for specified size of the covered set and a specified machine.

The present pilot project in this virgin field has an objective of verifying some conjectures. It establishes some principles of constructing, formatting, and storing a large data base in dictionary form. It develops programs for displaying, handling, and modifying such a data base. The paper offers an example how a conceptually continuous operation on large amounts of data can be reduced to operating on a fraction of the whole data base at a time by successive small increments of time. We finally demonstrate the feasibility of solving lexicometric problems on the computer and, at the same time, show the cost involved in doing such work in terms of both human effort and machine time.

We describe the program that accomplishes the above tasks,

and the results that were obtained in using an existing dictionary of computer terminology of more than 1,800 entries. The effort required was considerable: 6 man-month's work and about 14 hours of CDC 6400 computer time. Programming was done in SLIP/AMPPL-II, a list processing and associative memory plus parallel processing language package enbedded in FORTRAN IV.

TABLE OF CONTENTS

INTRODUCTION

Since the early days of electronic computing, two kinds of associations have existed between computers and dictionaries: either the computer uses, for various purposes, a stored dictionary of some sort (lexicon, vocabulary, glossary, thesaurus) or the computer is employed for constructing and analyzing a dictionary. The latter activity was given a strong impetus in the late 1950's by the formation of the Centre d'Etudes du Vocabulaire Francais and its publication, the Cahiers de Lexicologie. Thus lexicography was among the first non-mathematical disciplines to make use of the symbol manipulating capability of computers.

While formal theories of syntax have been successful in describing the rules of grammatical acceptability of natural language utterances, the study of meaning, usually called semantics, has not yet produced a theory of the semantic structure of languages, based on observation and analysis. It is beyond the scope of this paper to discuss, even superficially, the various viewpoints concerned with the concept of meaning. One of us, Viil (1974), has, however, compiled a reasonably exhaustive critical survey of the relevant literature.

For the purposes of this work, it suffices to present the following categories of meaning, as set out by Longyear (1971):

1. Logical meaning applies to such attempts to deal with meaning as symbolic logic and mathematics. The meanings with which the signals of such systems correlate are unique outside-world referents or unique meanings within the logical system that eventually have outside-world referents.

2. General-semantic meanings are also unique in their reference to outside world, but the semanticists are less stringent in scope than the logicians. Nevertheless, their scope is an idealized language, much more limited than ordinary language.

3. Communication-theory meaning is equivalent to the amount of information that can be transmitted per unit time in a communication system.

4. Lexicographical meaning is that of "words," and the outside-world reference is what we ordinarily call "meaning."

5. Psychological meaning has so great a scope that the part involving ordinary language becomes nearly trivial. It encompasses overt or covert behavior of any organism as responses to stimuli.

6. Word-mind meaning has the scope equivalent to that of ordinary language. The "words" here are linguistic structures, but the "meanings" are ideas, mental states, and

conceptual categories. To ordinary meanings (in the lexical sense) here correspond signals by which mental states are ascertained.

7. Linguistic meaning refers to signals as the pieces out of which language is made, i.e. microlinguistic, phonological, and syntactic signals.

In the framework of our particular topic we shall be mainly concerned with categories 4 and 7.

According to Weinreich (1966), unilingual defining dictionaries appear to be based on a model that assumes a distinction between meaning proper (signification, comprehension, intension) and the thing meant by a sign (denotation, reference, extension). On the basis of what is meant by a sign, Osgood, Suci, and Tannenbaum (1957) distinguish three kinds of meaning.

1. Pragmatical (sociological) meaning: the relation of signs to situations and behaviors.

2. Syntactical (linguistic) meaning: the relation of signs to other signs.

3. Semantical meaning: the relation of signs to their significates. It is easy to see that these classes are in correspondence with Longyear's three layers in category 7.

Homing onto our primary target, we may now restrict our interests somewhat further and concentrate on the two last classes of meaning, known under various designations but, by the majority of writers, distinguished as structural meaning and lexical meaning.

Mackey (1965) finds structural meanings in (1) structure words, (2) inflectional forms, and (3) types of word order. Examples of structure words are articles and prepositions, and these, he insists, although often called meaningless or empty, may have a large number of meanings. Similarly, the inflectional forms, such as the genitive case and present tense, may have a number of meanings, and so may some types of word order. Lexical meanings, on the other hand, refer to the meanings of the content words, in which the differences in meaning are most easily seen.

In Russell's view (1967) the structure words, such as "than," "or," "however," have meaning only in a suitable verbal context and cannot stand alone. The content words, which he calls object words, such as proper names, class names of animals, names of colors, do not presuppose other words and can be used in isolation. Their meaning is learnt by confrontation with objects that are what they mean or instances of what they mean. As soon as the association between an object word and what it means has been established by the learner's hearing, if frequently pronounced in the presence of the object, the word is understood also in the absence of the object. This explanation, of course,

excludes words that denote abstract entities, which are not object-like and usually cannot have a "presence." It also denies that every structure word inherently denotes one or a few definite relationships even in isolation. If this were not so, one could not understand what kind of relationship it designates if used in a context.

Lyons (1969), quite sensibly, distinguishes between three different kinds of structural, or grammatical meaning.

1. The meaning of grammatical items, such as prepositions and conjunctions.

2. The meaning of grammatical functions, such as subject and object, i.e. syntactical relations.

3. The meaning associated with notions such as declarative, interrogative, imperative, i.e. syntactical types.

He further rightly observes that grammatical items belong to closed sets, which have a fixed, small membership, e.g. personal pronouns. Lexical items, on the other hand belong to open sets, which have an unrestricted, large membership, e.g. nouns. Moreover, lexical items have both lexical (material) and grammatical meaning whereas grammatical items have only grammatical meaning.

In our work, the distinction between structure words and contents words is essential. This fact is clearly seen in the preparation of the dictionary used for our experiments.

SOME PROBLEMS OF LEXICAL RELATEDNESS

1. Polysemy and Homonymy

While the problem of meaning is complex in itself, the difficulty increases by another order of magnitude if one has to deal with words of many meanings or different words with different meanings that have identical spellings or pronounciations. And the decision as to whether a given case represents one polysemous word or two (or more) homonyms is far from being well defined.

The separation can be based on morphological criteria. First of all, two graphematically identical word forms with different meanings are regarded as homographs and separated if they display a phonematic difference or if they belong to different word classes. They are also homographs even if they belong to the same word class but possess different inflection systems. Otherwise, they represent the same word. More than one meaning of one word constitutes a case of polysemy. In contrast with such diversified meanings of one word, we talk about homonymy, in which case two words have by chance acquired the same external

appearance. A distinction between the two can only be made, if at all, on the basis of the historical origin of the words involved. Direct, transferred and specialized senses of a word can be listed along one dimension of meaning, dominant and basic senses represent certain measures along another dimension.

Another concept is semantic depletion, in which case the word occurs in scores of expressions. Here, the verbal or situational context adds substantially to the meaning of the word in question. With polysemy, however, the context eliminates those senses of the word that do not apply and thereby disambiguates the polysemous word. It is, therefore, important from the lexicographical point of view to distinguish between the degrees of interaction between the context and the meaning of individual words:

(a) in case of weak influence, we talk about autosemantic or semantically autonomous words;

(b) a strong influence performs a disambiguation of polysemous or homonymous words;

(c) the context defines the meaning of synsemantic or semantically depleted words.

Needless to say that the above, as innumerable other, decisions must often be based on subjective criteria. Finally,

it could be noted that, in exceptional cases, even the immediate context cannot resolve the ambiguity, and two or more interpretations are acceptable. This phenomenon is the pathology of language.

2. Synonymy

It is clear even to the casual observer that total interchangeability in all contexts, and identity in both cognitive and emotive senses, of two lexical units (words, in the simplest case) are not possible in general. The semantic relationship between synonymy is based on and measured by a level of similarity.

Rather than distinguishing between the "meaning" and the "usage" of a word, one should assume the view that the former is the sum total of the possibilities of the latter. This is basically what justifies the existence of any monolingual (and, possibly, bilingual) dictionary.

The entries in the dictionaries we are concerned with are both words (the interpretation and definition of which units are less than clear-cut) and multi-word lexical units. The two are of the same standing and function, and they will be treated identically.

3. Definitions

Definition is the most fundamental concept associated with dictionaries. We shall be concerned with both classical Aristotelian definitions, based on "class" and "characteristics", and operational definitions which use sentential generative terms. In fact, it is often difficult or impossible to separate equivalence or paraphrase definitions, on one hand, and those that are process-oriented reproductions, on the other.

In general, the lexical meaning can be rendered by four basic instruments and their various combinations:

(a) the lexicographic definition enumerates the most important features of the lexical unit being defined, in the simplest possible terms;

(b) qualified synonyms provide a system of semantically most related words;

(c) exemplification puts the defined unit in functional combination with other units;

(d) a gloss is an explanator or descriptive comment related to the dictionary entry; it may also state similarities to and differences from other entries.

ASPECTS OF THE SCIENCE OF DICTIONARY

1. General Concepts

Although definitions abound, a reasonable distinction seems to be to say that the semantic description of individual terms, the inventory of words is the customary province of lexicography whereas lexicology refers to the study of the lexical material, of the recurrent patterns of semantic relationships, and of any formal devices, such as phonological and grammatical systems, that generate the latter.

To construct a dictionary of a given size, one could choose the entries on the basis of their frequency of occurrence or in relying on some measure of utility that is vaguely tied to the semantic generality of the candidates. No solution is perfect or even uniformly useful over the whole dictionary.

Even the arrangement of meanings of a given entry is moot. We talk about logical, historical and empirical orders. (The latter starts with the common and current usage followed by obsolete, colloquial, provincial, slang and technical meanings.)

We can differentiate between encyclopedic and linguistic dictionaries. The latter are primarily concerned with the lexical units of the language and all their linguistic properties. The former, on the other hand, give information

about some component of the extralinguistic world. Our work derives its data base from an encyclopedic dictionary. It should be noted that the highly polysemous nature of the entries in a linguistic dictionary would have constituted an additional complication in this pilot project, which has now been avoided without affecting the general validity of the results.

We propose to introduce the term lexicometry to designate the discipline which investigates and analyzes the quantitative aspects of dictionaries, the vocabulary of a language and various subsets of the latter. Lexicometry would count, weigh and measure, and express the results in statistical and mathematical terms. Many such studies are widely known. Such is the one reported by Guiraud (1959):

> The most frequent words are:
> (a) the shortest,
> (b) the oldest,
> (c) the morphologically simplest,
> (d) the semantically most extended, i.e. possessing the greatest number of meanings.
>
> As to the measure of frequency,
> the first 100 words cover 60% of an "average" text,
> " " 1000 " " 85% " " " ,
> " " 4000 " " 97.5% " " " .

Thus the remaining X(?) thousand words cover only 2.5% of the text. However, from an information theoretic point of view,

the first 100 words comprise 30% of the information,
" " 1000 " " 50% " " "
" " 4000 " " 70% " " "

Consequently, rare words convey a great deal of information. We could say that a frequent word is most useful in the aggregate, and a rare word in a particular case.

Other studies in glottochronology concern themselves with the rate of change in language and in basic vocabulary. Further, distribution of the frequencies of occurrence with or without reference to any particular vocabulary has also been studied.

Finding relations of the above kind is not just an academic exercise to satisfy the curiosity of a few linguists, but these relationships may have various practical applications. For example, Maas (1972) asserts that the knowledge of a functional relation between the length of a text and the size of the vocabulary used in it would be desirable in order to estimate the effort needed for extension of a machine dictionary or in comparison of vocabulary contents of texts of different lengths. In the latter case, one can standardize or normalize the texts under investigation by reducing them to a common minimal length through computational methods and then compare the resulting vocabulary volumes.

Let V be the number of elements (words) in a text and N the length of the text. Then we surmise, says Maas, a functional relationship to exist between N and V:

$$V = f(N)$$

Muller (1964) reported a relation between V and N such that the ratio of their logarithms is constant:

$$\frac{\log N}{\log V} = \alpha, \text{or } V^{\alpha} = N, \quad \text{or,}$$
$$\text{if we set } \frac{1}{\alpha} = k, \quad V = N^{k}.$$

Since the vocabulary of a language, however, is supposed to be restricted, so argues Maas, the existence of a limiting value is to be postulated:

$$V_0 = \lim_{N \to \infty} f(N)$$

As the derivative of f at a given value of N represents the relative increase in V, it is to be stated that $f'(N)$ approaches 0 with increasing N.

The derivative of f at the point 1 is assumed to be 1 because a text of length 1 has a vocabulary consisting of one word, hence

$$f'(1) = 1$$

Therefore f' is a function that decreases monotonically from 1 to 0.

As a consequence of the above speculations, in the expression $V = N^k$, k cannot be constant.

Statistical investigations of the dramas by Corneille have resulted in the relationship

$$\log \frac{1}{k} = 0.0137.(\log N)^{1/3}.$$

Thus, if N is given, k can be determined, and V can be calculated from

$$k = \frac{\log V}{\log N}.$$

Another noteworthy concept is that of repetition factor:

$$R = \frac{N}{V},$$

which shows how often a word has occurred in a text on the average.

The following relationship has been determined:

$$\log R = (0.179 \log N + 0.026)^2,$$

which displays a very good agreement with reality.

No single empirical law seems to exist between N and V for all N.

2. The Problem of Coverage

We are now coming close to the core subject matter of this paper. Mackey (1965) states that

> "The coverage or covering capacity of an item is the number of things one can say with it. It can be measured by the number of other items which it can displace."

According to him, words can displace other words by four means: (1) inclusion, (2) extension, (3) combination, and (4) definition.

1. A word that already includes the meaning of other words can be used instead of these (e.g., seat includes chair, bench, stool, and place).

2. Words the meanings of which are easily extended metaphorically can be used to eliminate others (e.g., tributary of a river can be covered by branch or arm).

3. Certain simple words can displace others by combining either together or with simple word endings (e.g., news + paper + man = journalist; hand + book = manual).

4. Certain words can be replaced by simple definition (e.g., breakfast can be defined as morning meal; pony as small horse).

As an example of the application of the above principle, in the derivation of Basic English (by definition), the language was first reduced to 7500 words, and, by redefinition, cut down to 1500. These were further reduced to the eventual 850 by a technique of "panoptic" definition (eliminate each word on the grounds that it is some sort of modification of other words, e.g. a modification in time, number, or size).

Basic English, which was founded essentially on the principle of coverage, was a conscious reaction against the over-application of the principle of frequency in selection. For Ogden (1933), it was not the frequency of a word which makes it useful, it was its usefulness which makes it frequent.

In the following part of this section, we attempt to present some of the salient points of Savard (1970).

The vocabulary indices most widely known today are those of frequency, of distribution, and of availability. But these are

not sufficient to select words for a restricted vocabulary for the purpose of teaching a foreign language, such as French, to beginners.

An objective criterion is lexical valence. It would allow

1. to obtain a novel principle of vocabulary selection,

2. to assist the investigators in setting up a base vocabulary for French,

3. to provide a usable definition, combination, inclusion, and extension vocabulary,

4. to correct all the already existing scales of French vocabulary,

5. to provide a valid working tool for the analysis of teaching material.

The valence problem is a problem of verbal economy. What he calls valence is the fundamental capability of a word to be substituted for another word. It is Mackey's coverage that he renders as valence.

Like Mackey (1965), he maintains that the substitution of one word for another can be made by virtue of four criteria: (1) definition, (2) inclusion, (3) combination, (4) extension.

Definition has already been discussed previously.

Linguists do not talk specifically about inclusion; rather, they deal with synonymy or lexical parallelism. Synonyms are words that have nearly the same meaning, e.g. lieu and endroit. For Savard, the basic criterion that permits to establish a series of synonyms is the possibility of substituting one term for another.

One of the simplest among all the procedures of vocabulary enrichment consists of joining two or more words in order to make compound words. The principle of combination appears as another phenomenon common to all languages.

It is not necessary that the number of simple words be unbounded because almost all verbs have a potential of undetermined sense, and so do the adjectives. A word is said to have more or less extension according to whether it can "cover" a more or less great number of fully or partially different notions.

Polysemy is the exact opposite of synonymy. Polysemy becomes complicated due to the phenomenon of homonymy. Polysemy and

homonymy constitute two very rich sources of lexical economy. Together they form Savard's last criterion of lexical valence--the semantic extension.

Although the valence itself has never been mathematically measured and although there exists no scientific means of showing its existence, it has nevertheless been proven that four formal procedures of lexical economy permit to replace certain words by other words, and that is what Savard calls lexical valence.

The postulated existence hypothesis of lexical valence leads to the calculation of a global index of valence for every word.

To evaluate the power of <u>definition</u> of a word, one inspects, in the dictionary, each element of the general list and counts how many times a word enters into the definition of another.

To measure the power of <u>combination</u> of a lexical unit, one inspects in the dictionary all the compound words joined by a hyphen, all the Gallicisms (in English, these would be Anglicisms) and, in general, all the word groups.

With a view of appraising the power of <u>inclusion</u>, one inspects the units of the general list in two synonym dictionaries and takes the higher number. The number of synonyms that possess a word constitutes a measure of the number of words for which it can be substituted.

To measure the power of semantic extension, one inspects each of the elements of the general list in the dictionary and counts the number of meanings given by the author to such a word in the list. The number of meanings of a word is considered as a measure of its power of semantic extension.

The global index of lexical valence is the sum of the four normalized counts. The two criteria having the highest correlation are definition and combination.

In the beginning of the study, it was assumed that the four variables were entirely independent of each other. The results of a factor analysis indicate that they are not completely so. A factor rotation shows, however, that the variables are sufficiently independent to make it necessary to retain the four criteria of lexical valence.

A comparison of the rank of the first 40 content words on the valence scale with the same words on the frequency list allows to frame a hypothesis that the correlation between valence and frequency would be rather weak. A more complete study would show without doubt that we have there two very different selection principles.

In conclusion, it can be stated with confidence that the measure of valence is no less valid than that of frequency, distribution, and availability. These concepts will eventually lead to more efficient dictionaries with respect to precision, compactness and lexical economy.

ON LEXICOMETRIC RELATIONSHIPS AMONG THE SIZE OF DEFINING SET, THE SIZE OF DEFINED SET AND THE MAXIMUM LENGTH OF DEFINITIONS

1. Some Measures of Coverage

A dictionary may be considered efficient and economical if it uses a reasonably small set of words to define a relatively large set of entries. We have, however, a very vague idea about what size vocabulary is needed to cover a given number of dictionary entries. (The related problem of circular definitions seems to have to wait for a computer solution.)

It is known, for example, that Basic English, Ogden (1933), involves a list of 850 English words and 50 international words, which were eventually used to define the 20,000 English words of Basic English Dictionary. This gives a ratio of the number of covering words to that of defined words of 0.045.

West studied the problem of what constitutes a simple definition and established a minimum defining vocabulary of 1,490 words. The meaning of some 18,000 words and 6,000 idioms, i.e. about 24,000 expressions, was explained exclusively by these 1,490 words, which were not defined themselves. The results were published in 1961 as The New Method English Dictionary by M.P. West and J. G. Endicott. The corresponding size ratio here is 0.062.

The above roughly indicates that a set of about 1,000 words

can define a set of about 20 times that size, but in general the behavior of these variables has not been investigated and is not known in any detail.

One of us, in Findler (1970), has formulated the problem in definite terms. Three variables were considered: (1) the covered set S of size v_S, (2) the Covering set R of size v_R, and (3) the maximum definition length N, such that each word in S can be defined by at most N ordered words from R. The task is to find:

(a) v_R as a function of v_S at different values of N as a parameter, and

(b) v_R as a function of N at different values of v_S as a parameter.

Using the terminology of increment ratio for $\Delta v_R/\Delta v_S$ and size ratio for v_R/v_S, it was postulated for case (a) that

* the increment ratio is, in general, less than one[2],

* the increment ratio, in general, decreases as v_S increases[2],

* for large values of N, v_R asymptotically approaches a limiting value as v_S increases,

* the increment ratio will never exceed the size ratio.

---

[2] An exception to this rule would occur in a dictionary system, which does not treat homonyms as individual entries, every time a new word with many homonyms is introduced into the Covered Set.

It was further assumed that for N=1, the covering set and the covered set are of the same size, i.e. both the increment ratio and the size ratio equal one. We must now correct this statement because not every word is defined by itself only. If a new word is introduced that already has a synonym in the covering set, it will be defined by that synonym. Then the increment ratio is 0 and the size ratio become less than 1.

For the second case, (b), it is postulated that

* v_R monotonically decreases as N increases,

* for any fixed v_S value, v_R asymptotically approaches a lower limit as N increases without bound.

It was finally pointed out that v_R should be small to minimize storage requirements, and N should be small to minimize processing time and output volume. A compromise on these conflicting requirements is needed. The ultimate question is: "What are the optimum v_R and N values for $a^{given}_{\Lambda\, v_S}$ for certain computer applications on a machine with a given cost structure?"

It is reasonable to assume that the behavior of the three variables and therefore the answer to the last question will largely depend on the semantic index of the elements of the covered set and on the lexical valence of the elements of the covering set. The latter implies that, for an efficient and economical dictionary, the elements of the covering set must be chosen from the available vocabulary on the basis of a careful

analysis. As research aimed at these goals is pratically nonexistent, it is safe to assume that most of the existing dictionaries are suboptimal. Work in this area will be useful, challenging, and rewarding, but the investigators must be prepared to spend a considerable amount of time and effort on it. So much the more as the entire problem complex outlined in the preceding parts will directly or indirectly enter into such investigations.

The project described here is only a small beginning. It was originally intended to complete the investigation of both cases, (a) and (b), defined above. In view of the effort needed, in terms of human and machine time, only the first part is accomplished at the time of writing this report. Appendix II contains the design of the program for case (b).

2. Construction of the Data Base

The data base was not derived from a text but was based on an existing dictionary of computer terminology, Chandor (1970). A derivation from a text, if used, should be automatic and would constitute a large-scale programming project in its own right. In creating the data base, it was attempted to keep its structure simple and uniform without sacrificing its general validity. It was tried to avoid problems that would introduce distracting complications, from both theoretical and practical point of view, into the subsequent operations. All this led to the selection

and construction principles outlined below.

Terms with excessively long definitions were avoided, i.e. definitions were held reasonably short. It was found that limiting the maximum definition length to 22 lexical units did not unduly restrict the selection. In some cases too long definitions were shortened by leaving out redundant words, glosses, or explanatory notes.

Every element of the covered set was considered a lexical item, regardless of whether the original dictionary entry consisted of one, two, or more words. For programming convenience every word was coded as a string of no more than 10 symbols. Thus accumulator was represented as ACUMULATOR, absolute address appeared as ABSADDRESS, and absolute value computer as ABSVALCOMP.

Polysemous terms were avoided. If such a term was used, only its dominant meaning was recorded. In the data-base dictionary, then, each entry (element of the covered set) has only one meaning and one definition.

Terms used in the definitions (elements of the covering set) were also considered to be lexical items, i.e. original multiword terms appear as a single element, and every element is represented as a string of no more than 10 symbols.

All terms occurring in the definitions are themselves defined, i.e. each element of the covering set appears also in the covered set. This principle implies that there is a set of words each element of which is defined by itself. Such a set may be called the basic vocabulary, consisting of words the meanings of which the user of the dictionary is supposed to know in order to use the dictionary. As in this particular case, the dictionary is one of computer terms and the basic vocabulary contains the nontechnical words used in the definitions of the technical terms.

In the definitions, a definite distinction was made between content words and function words, also called operators. The latter were not included in the covering set nor were they counted in determining the definition length. Hence, the covering set consists only of content words.

The set of function words is defined rather broadly. It contains a wide variety of expressions that do not directly contribute anything to the content of the definition but only indicate grammatical and logical relationships between the words that form the content. It includes:

1) prepositions, e.g. of, in, to;

2) conjunctions, e.g. and, or, if;

3) the relative pronoun which;

4) combinations of preposition and relative pronoun, e.g. in which, to which, by which;

5) present participles equivalent to a preposition, e.g. using, containing, representing;

6) combinations of participle and preposition, e.g. consisting of, opposed to, applied to;

7) combinations of adjective and preposition, e.g. capable of, exclusive of, equal to;

8) combinations of noun and preposition, e.g. part of, set of, number of;

9) combinations of preposition, noun, and preposition, e.g. in terms of, by means of, in the form of;

10) prepositional phrases associated with a following infinitive, e.g. used to, necessary to, in order to;

11) other frequently used purely functional expressions, e.g. for example, namely, known as.

Actually, the function words were replaced by code numbers in the dictionary. The code numbers were assigned consecutively as the function words were needed during the construction of the data base so that the order is purely random. A complete list of the 121 function words used, together with their code numbers, is given in Table I.

INSERT TABLE I ABOUT HERE

---

1. is equivalent to
2. of
3. in
4. in terms of
5. using
6. and
7. which
8. in which
9. between
10. to
11. or
12. from
13. used to
14. necessary to
15. part of
16. consisting of
17. containing
18. capable of
19. by means of
20. opposed to
21. when
22. on
23. so that
24. in order to
25. exclusive of
26. for
27. pertaining to
62. if
63. among
64. by
65. namely
66. related to
67. concerned with
68. based on
69. constituting
70. resulting from
71. set of
72. including
73. followed by
74. provided by
75. developed by
76. assigned to
77. referred to
78. on which
79. used as
80. in the form of
81. from which
82. into which
83. number of
84. less
85. defining
86. known as
87. performing
88. performed by

28. under
29. as
30. such as
31. equal to
32. into
33. with
34. according to
35. applied to
36. depending on
37. to which
38. whose
39. obtained by
40. inherent in
41. through
42. during
43. where
44. during which
45. out of
46. at
47. by which
48. used in
49. without
50. caused by
51. over
52. not
53. but
54. extended to

89. independent of
90. chosen by
91. for which
92. at which
93. whether
94. used by
95. about
96. before
97. per
98. having
99. formed by
100. around
101. after
102. since
103. against
104. until
105. whereupon
106. except
107. determined by
108. over which
109. in relation to
110. belonging to
111. corresponding to
112. due to
113. required for
114. type of
115. across

55. so as to
56. for example
57. represented by
58. along which
59. representing
60. against which
61. similar to
116. because
117. designed to
118. indicating
119. produced by
120. outside
121. towards

TABLE I

List of Function Words

The original definitions were somewhat simplified and standardized. In this process, articles were omitted (many languages do very well without them). On the other hand, implicit relationships were made explicit. A few examples shall serve as illustrations, with the function words (in parentheses) inserted explicitly instead of their code numbers.

Original dictionary entry:

aberration A defect in the electronic lens system of a cathode ray tube.

Definition in the data base:

DEFECT (in) SYSTEM (of) ELECTRONIC LENS (of) CATHRAYTUB

Note that electronic lens system" (should be: electronic-lens system) means "system of electronic lens" (as opposed to "electronic system of lens"), and this relationship is made explicit. Note also that "cathode ray tube" is a single lexical item.

Nouns are represented in singular, thus avoiding another dictionary entry for plural or, what would be worse, programming a "grammar." Likewise, finite verb forms are represented in third person plural present indicative active. Avoiding the third person singular eliminates another dictionary entry, and avoiding the passive voice eliminates a great many participles, which otherwise would have had to be entered. Of course, present and

past participles (the former identical to gerund in form) could not always be avoided and had to be entered in the dictionary where needed. Auxiliary verbs were automatically eliminated by avoiding compound tenses and the passive voice. Finally, "to do" associated with negation was simply omitted.

Original:

absolute coding Program instructions which have been written in absolute code, and do not require further processing before being intelligible to the computer.

Data-base entry: ABSOCODING

Definition:

PROGRAM INSTRUCTIO (which) ONE WRITE (in) ABSOLUCODE (and which not) REQUIRE FURTHER PROCESSING (before) INTELIGIBL (to) COMPUTER

Note that the first predicate in the relative clause, third person plural perfect indicative passive, is represented by the singular indefinite pronoun "one" as subject, followed by the standard plural active verb. The auxiliary "do" has been omitted and the negation is represented by a function word. The virtually redundant "being" has also been left out. In general, the copula is omitted (some languages do very well without it).

Original:

analytical function generator A function generator in which the function is a physical law. Also known as natural law function

generator, natural function generator.

Data-base entry: ANLYTFNGEN

Definition:

FUNCGENRTR (in which) FUNCTION PHYSICAL LAW

Note also the omission of the gloss "Also known as

The stylized definitions are easily understandable even to human readers as the printout of the dictionary demonstrates.

The data base was constructed by selecting the first entry, then entering all the lexical items in its definition, subsequently entering all the lexical items in the definitions of these etc. Words that were not defined in the original dictionary were entered and defined by themselves; they constitute the basic vocabulary. This procedure was continued until everything was defined, i.e, until all the terms in the covering set were also in the covered set. Then the next entry was selected from the dictionary, and the above process was repeated.

It had been tentatively intended to compile a covered set of about 1,000 lexical items. When this number was reached, a rough pencil-and-paper check indicated that the size ratio was about 0.91 at that point. It was then decided that the data base sould be somewhat larger to show the relationships under investigation

more perceptibly, and more words were added.

When the size ratio had decreased to about 0.79, the construction of the data base was concluded as processing difficulties were anticipated with too large a data volume. At that point the data-base dictionary had precisely 1,856 entries (as was later verified by the program). This was considered to be a satisfactory compromise.

The dictionary was arranged in the form of a SLIP list, Findler et al. (1971). Every entry (element of the covered set) occupies four cells in this list: (1) entry word (in A10 format), (2) definition length (an integer), (3) type of entry (an integer), (4) sublist name.

Three types of entries were distinguished for programming convenience:

1) code 0 indicates that the entry itself is not used in any definition, i.e. it occurs only in the covered set and not in the covering set;

2) code 1 indicates that the entry occurs in both sets and is not an element of the basic vocabulary;

3) code 2 indicates that the entry is defined by itself, i.e. it belongs to the basic vocabulary.

The sublist, the name of which is in the fourth cell for every entry in the main list, contains the definition. This

arrangement conveniently separates the entry words from those in the definitions.

A cell in this second level contains either a word (in A10 format), i.e. an element of the covering set, or a sublist name. The codes for function words (integers) are contained in the cells in the third level. This arrangement is convenient for bypassing the function words in processing when they are not needed. A typical dictionary entry is illustrated in Figure 1.

INSERT FIGURE 1 ABOUT HERE

The fact that every dictionary entry owns a sublist is practical in another respect: useful information about the entry can be collected and deposited in a description list associated with the sublist. For example, if it were desired to evaluate the definition component of the lexical valence of each lexical item, a program could be developed that counts how many times a particular item occurs in the definition of other items and stores this information in the description list created for that item. Investigations of this nature will be done at a future date.

The program developed for processing all the necessary information is rather complex. Since many of its organizational characteristics may be of fairly general interest to those who wish to engage in lexicometric studies, a brief description is

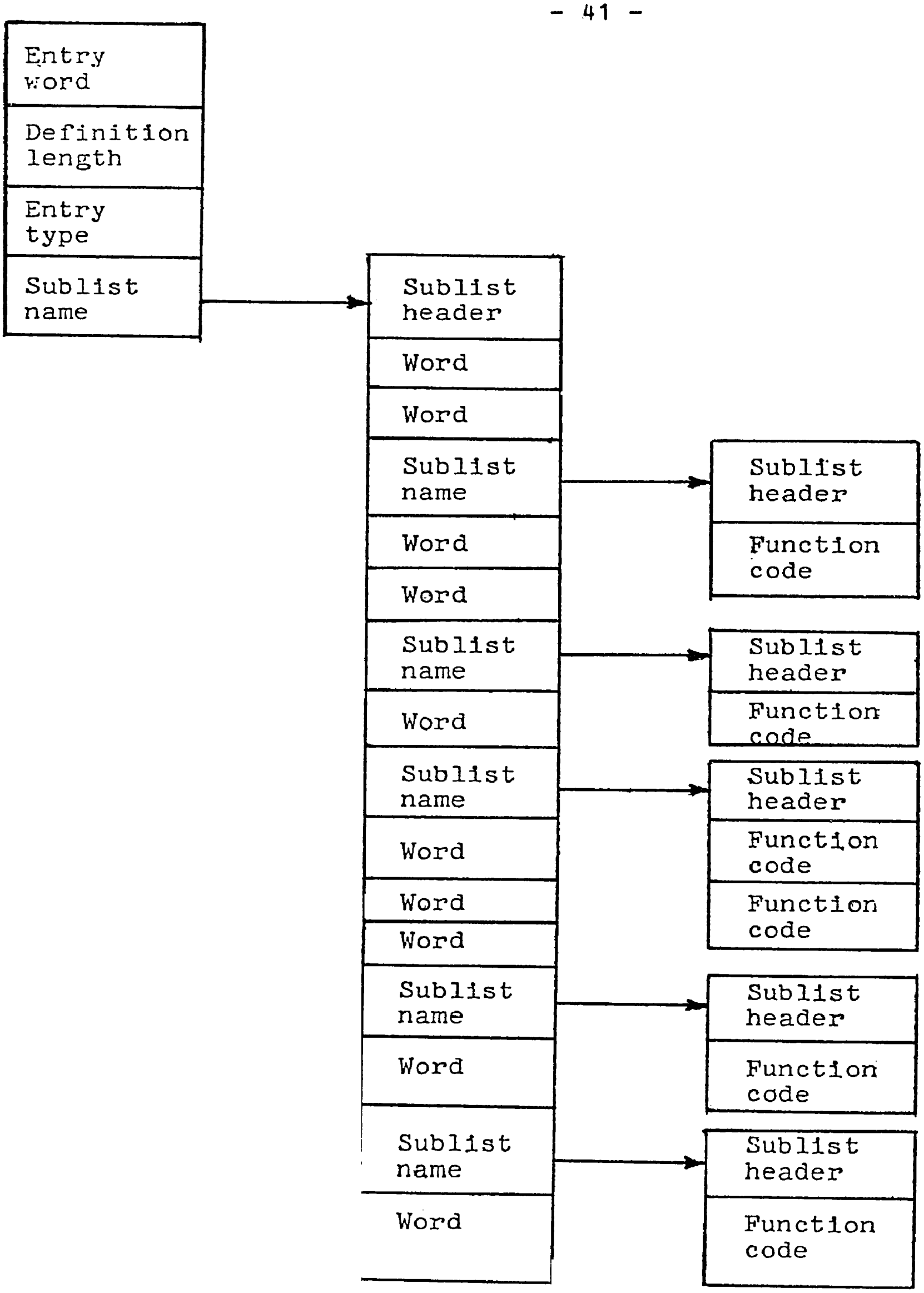


Fig. 1.—A representative entry in the data-base dictionary

given in Appendix I.

3. The Results of the Computations.

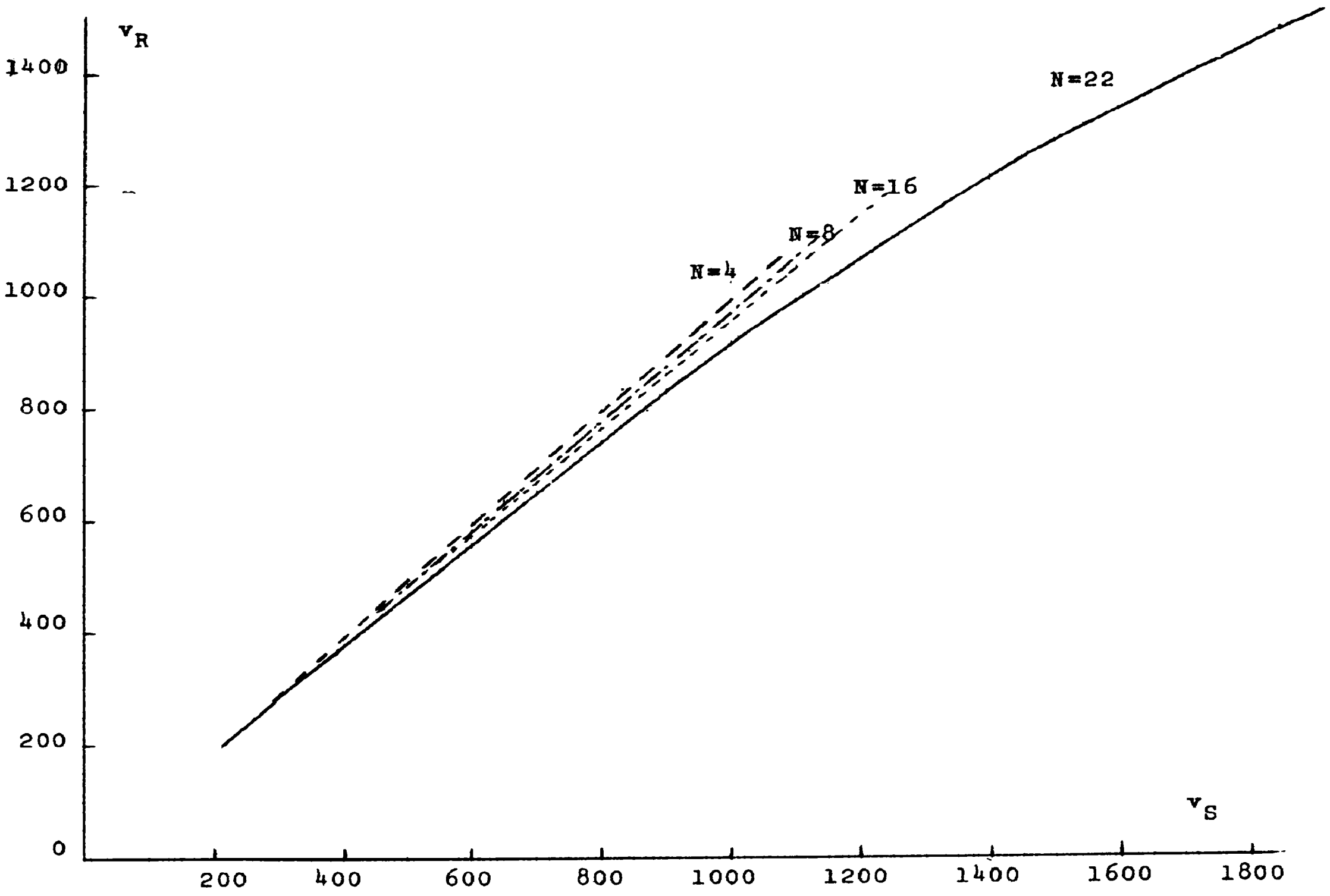
The relationships between the size of the covering set v_R and that of the covered set v_S are summarized in Table II. The table lists the size of both sets, the size ratio, the increment of either set, and the increment ratio for four values of N. Figure 2 presents v_R as a function of v_S, with N as a parameter, in graphical form.

INSERT TABLE II AND FIGURE 2 ABOUT HERE

The table shows that, in general, the increment ratio is less than 1, except for one case, to which we shall return below. In the meantime we note that, for the full dictionary, the table definitely verifies the assumption that the increment ratio decreases with increasing v_S. This, however, does not seem to be true for the reduced dictionary. In fact, for all three cases of the latter, the ratio tends to increase with increasing v_S. Therefore the single occurrence of the value 1 is plainly a random event as the ratio is very close to 1 at the largest v_S value also in the two other cases. The sequence of values is evidently approaching unity.

This somewhat unexpected, though not particularly surprising,

Fig. 2. - Variation of the size of Covering Set with that of Covered Set



v_S	v_R	$\frac{v_R}{v_S}$	Δv_S	Δv_R	$\frac{\Delta v_R}{\Delta v_S}$
			N = 22		
574	573	0.998	200	165	0.825
774	738	0.955	202	157	0.777
976	895	0.916	202	151	0.748
1178	1046	0.888	200	143	0.715
1378	1189	0.862	202	123	0.609
1580	1312	0.832	201	117	0.582
1781	1429	0.800	75	35	0.466
1856	1464	0.790			
			N = 16		
201	189	0.940	156	150	0.962
357	339	0.950	100	94	0.940
457	433	0.947	129	125	0.969
536	558	0.953	198	194	0.990
784	752	0.959	125	123	0.984
909	875	0.963	129	126	0.984
1037	1001	0.965	185	195	1.000
1222	1186	0.971			

N = 8					
287	275	0.958	141	136	0.965
428	411	0.960	114	109	0.956
542	520	0.959	151	149	0.987
693	669	0.965	126	124	0.984
819	793	0.968	128	127	0.992
947	920	0.971	185	184	0.995
1132	1104	0.975			
N = 4					
239	234	0.979	131	129	0.985
370	363	0.981	294	292	0.993
664	655	0.986	383	382	0.997
1047	1037	0.990			

TABLE II

Covered-Covering Relationships

phenomenon is due to the combination of a number of circumstances. We are dealing with a specific technical dictionary. In such a dictionary, nontechnical, i.e. ordinary-language, words are not defined. However, a sizeable set of nontechnical words is necessary to define the technical terms. All the former, in our case, belong to the set of basic vocabulary and are defined by themselves. The result is an inordinate proportion of the set of basic words even in the full dictionary. A rough pencil check during the construction of the data base showed that the basic vocabulary forms about 0.55 of the entire covered set.

We recall that, in anticipation of this kind of difficulty, the function words were eliminated from the covering set, to begin with. If this had not been done, the situation would have been aggravated by an order of magnitude. To eliminate, or at least to alleviate this bias, a considerably larger data base should be used, which, as explained before, would have been beyond the scope of this pilot project.

Another, and more important, factor that contributes to the problem in question is the fact that our data-base dictionary was not derived from a text but constructed from another dictionary. This was done, as described earlier, by selecting entries starting from the beginning of the dictionary and stopping when the data base was of satisfactory size. As a result, while the basic vocabulary may be assumed to be uniformly distributed over the dictionary, the important content words, with longer definitions, are not. The selection of entries, in fact, was stopped at the letter H. Words beyond that point are there only because they happened to occur in definitions. Thus, at least the words that occur only in the covered set (and not in the covering set) are crowded toward the beginning of the dictionary.

What happened when the dictionary was reduced is now obvious. The weighty words with long definitions were eliminated but the entire basic vocabulary remained. This, of course, is quite appropriate and consistent with our principles. If, for example, the dictionary had been reduced to N = 1, virtually only the basic vocabulary would have been retained, and we should have obtained the postulated linear one-to-one relationship between $\underline{v_F}$ and $\underline{v_S}$. Nevertheless, this procedure enhances the proportion of the basic vocabulary, and the bias increases. As the technical words are relatively scarce in the last third of the dictionary to begin with, the situation gets worse, with the reduction, toward the end of the dictionary. This accounts for the increasing increment ratio. The last increment with N = 16 must

have consisted entirely of basic words, therefore the ratio of unity.

It is suggested that, for further investigation, a more complicated dictionary-reduction program be developed, which would compare all the basic words with all the remaining definitions and eliminate those that do not occur in any definition. Thus a basic word would occur in the dictionary only if it is needed in a definition, which was the case in the unreduced dictionary This way a more natural proportion between the basic words and others would be restored.

It is the same set of circumstances that also explains the fact that, in the reduced dictionary, the increment ratio almost consistently exceeds the size ratio. This, however, is not the case for the full dictionary, which definitely verifies the respective assumption in Findler (1970).

To demonstrate that v_R approaches an upper limit with increasing v_S for large N, a much larger dictionary would be needed. However, the curve in Figure 2 for N = 22 unmistakably shows a tendency in this direction.

There is, of course, another way of varying N: instead of reducing it, it could be increased, and certain words in the definitions could be replaced by their definitions. This would be a complicated procedure and difficult to control. If few such

replacements are made, $\overline{v}_R$ will not change appreciably. If many are made, some replacements tend to reintroduce precisely the words others try to eliminate. In any case, the result would be a set of awkward and unnatural definitions of erratic lengths. In order to use such a procedure, an efficient dictionary should first be compiled, with short definitions and well controlled covering set. The concept of lexical valence should be utilized, but this entails more research in this area. It would also get the researcher involved in the problem discussed in the preceding parts.

The curves for N = 16, N = 8, and N = 4 in Figure 2 all display the basic-vocabulary bias of the reduced dictionary. The last one very nearly approximates a one-to-one ratio. We must appreciate the fact that the 1,047 entries of the respective reduced dictionary contain about 1,000 basic words.

It is also to be noted that the full dictionary, with N = 22, in the region of $\overline{v}_S = 600$ requires a larger covering set than any of the reduced versions. This is understandable as we realize that the routine that computes the data points actually simulates, rather artificially, the construction of a dictionary from a source text. The full dictionary at that stage is close to encompassing the whole source, where complex technical terms are being defined, whereas the reduced versions, at the same value, are already in the area in which the basic vocabulary dominates.

The project has been informative in another respect, which is not unimportant: it has given an indication of the effort involved in this type of work. It has taken a total of about 14 hours of computer time. The development of the dictionary-display program and obtaining the printout was a matter of about 7 minutes and is therefore negligible. Of the 14 hours, about 3 were spent on dictionary reduction (three series of runs) and 11 on the analysis. Although some debugging had to be done, this was generally insignificant as compared to the total effort, so that nearly all the 14 hours has been useful running time.

It is also interesting that time seems to be very dependent on the volume of data being handled. Of the 11 hours, more than 9 were spent on running the full dictionary (N = 22) and about 1 hour on the reduced version of N = 16. Completing the running of the last two series (N = 8 and N = 4) tool together less than an hour of machine time.

In terms of human effort, the accomplishing of the project required about six man-months' work.

Finally, Appendix II contains a brief description of a plannned program that would investigate the relationship between the size of the covering set and the maximum definition length for fixed values of the covered set size.

ACKNOWLEDGEMENT

We wish to express our gratitude to the management of Penguin Books Ltd. for the permission to use their publication A Dictionary of Computers by A. Chandor as source for generating the data base of this project.

REFERENCES

Chandor, A. (1970) A Dictionary of Computers, Penguin Books, Harmondsworth, England.

Findler, N.V. (1970), Some conjectures in Computational Linguistics, Linguistics, No. 64, 5.

Findler, N.V., J.L. Pfaltz and H.J. Bernstein (1972), Four High Level Extensions of FORTRAN IV: SLIP, AMPPL-II, TREETRAN and SYMBOLANG, Spartan Books, New York.

Guiraud, P. (1959), Problemes et methodes de la statistique linguistique, Reidel, Dordrecht.

Longyear, C.R. (1971), Linguistically determined categories of meanings, Janua Linguarum, series practica, 92, Mouton, The Hague, Holland.

Lyons, J. (1969). Introduction to Theoretical Linguistics,

Cambridge University Press, Cambridge, England.

Maas, H.D. (1972), Über den Zusammenhang zwischen Wortschatzumfang und Länge eines Textes, Zeitschrift für Literaturwissenschaft und Linguistik, 2, No. 8, 73.

Mackey, W.F. (1965), Language Teaching Analysis, Indiana University Press, Bloomington.

Muller, C. (1964), Essai de statistique lexicale, Librairie Klincksieck, Paris.

Ogden, C.K. (1933), Basic English: An Introduction with Rules and Grammar, 4th ed., Kegan Paul, Trench, Trubner & Co., London, England.

Osgood, C.E., G.J. Suci and P.H. Tannenbaum (1957), The Measurement of Meaning, University of Illinois Press, Urbana, Illinois.

Russel, B. (1967), An Inquiry into Meaning and Truth, Penguin Books, Baltimore, Maryland.

Savard, J.G. (1970), La valence lexicale, Didier, Paris.

Viil, H. (1974), Some Lexicometric Properties of a Dictionary,

Unpublished M.S. Project at the State University of New York at Buffalo.

Weinreich, U. (1966), Explorations in semantic theory, in "Current Trends in Linguistics, Vol. III: Theoretical Foundations" (T.A. Sebeok, Ed.), pp. 395-477, Mouton, The Hague, Holland.

APPENDIX I

Program Development

The entire data base was first punched on cards to be inputted as a single list structure, with the dictionary entries alphabetically ordered. It was soon established that this arrangement by far exceeded run-time storage limitations (using a field length of $100{,}000_8$). Only about one fifth of the material could be accommodated at one time without exhausting the available space. Therefore the dictionary was split into five individual list structures, and the corresponding card images were stored on disk as five separate files. These were brought in, one at a time, for processing as needed. Because of space limitations, also processed data and intermediate results had to be put in external storage during run time and, of course, between runs, therefore more files had to be created as described later. Thus, a great deal of programming effort went into file manipulation.

The purpose of the first program, designated ANALEX, was simply to display the dictionary. It first reads the function words from the cards and stores them in the form of a 121x2 array. (The width of the array is 2 because many function words are longer than 10 characters.)

Using a function READLS, the program reads the dictionary and

stores it in the form of a list structure as described above. On this occasion, it also measures the space required for the dictionary. It was found that a field length of more than $235{,}600_8$ locations would be needed to accommodate the entire data base.

A subroutine called RITELS prints out the dictionary, specifying each entry by the definition in the form of at most 10 words to the line. The routine also checks the operator code numbers in the third-level sublists and replaces these in the printout by the appropriate function words from the array.

The dictionary was printed out in four separate runs as the dictionary was initially divided into four lists. Since the ANALEX program does no further processing and accumulates no new lists, no storage problems arose. It was not until later that it was established that a division into five parts was necessary to perform subsequent operations in the space available.

The first printouts were carefully examined for punching errors and omissions. Detected errors were corrected and the files were updated accordingly.

The actual working program is named COVSET. If the entire data base were one single list and if time were available indefinitely, this program would do the complete work in a single run. In this case, it would print a table of corresponding v_S

and v_R values for a given value of N, would reduce the value of N and print out another table, etc., and repeat this for all desired values of N.

This, of course, could not be done because, in the first place, only one of the five parts of the dictionary could be worked on at a time and, in the second place, the program had to be run in time increments of 600 s or less, which was the set time limit.

The principal routine in COVSET is called COVRNG, which computes the values of v_R for given values of v_S. Its simplified flow diagram is given in Figure 3.

INSERT FIGURE 3 ABOUT HERE

As the inherently continuous program cannot be run continuously, a few control variables are needed to provide criteria for interruption and to transfer information from one run to the next. These are read from cards in the beginning of the routine.

A reference value LSTREF is used to control the spacing of the recordings of v_S and v_R because too close spacing would introduce random irregularities into the otherwise smoothly changing tendency. The reference is automatically updated after every printout of the v_S and v_R values. During the analysis of

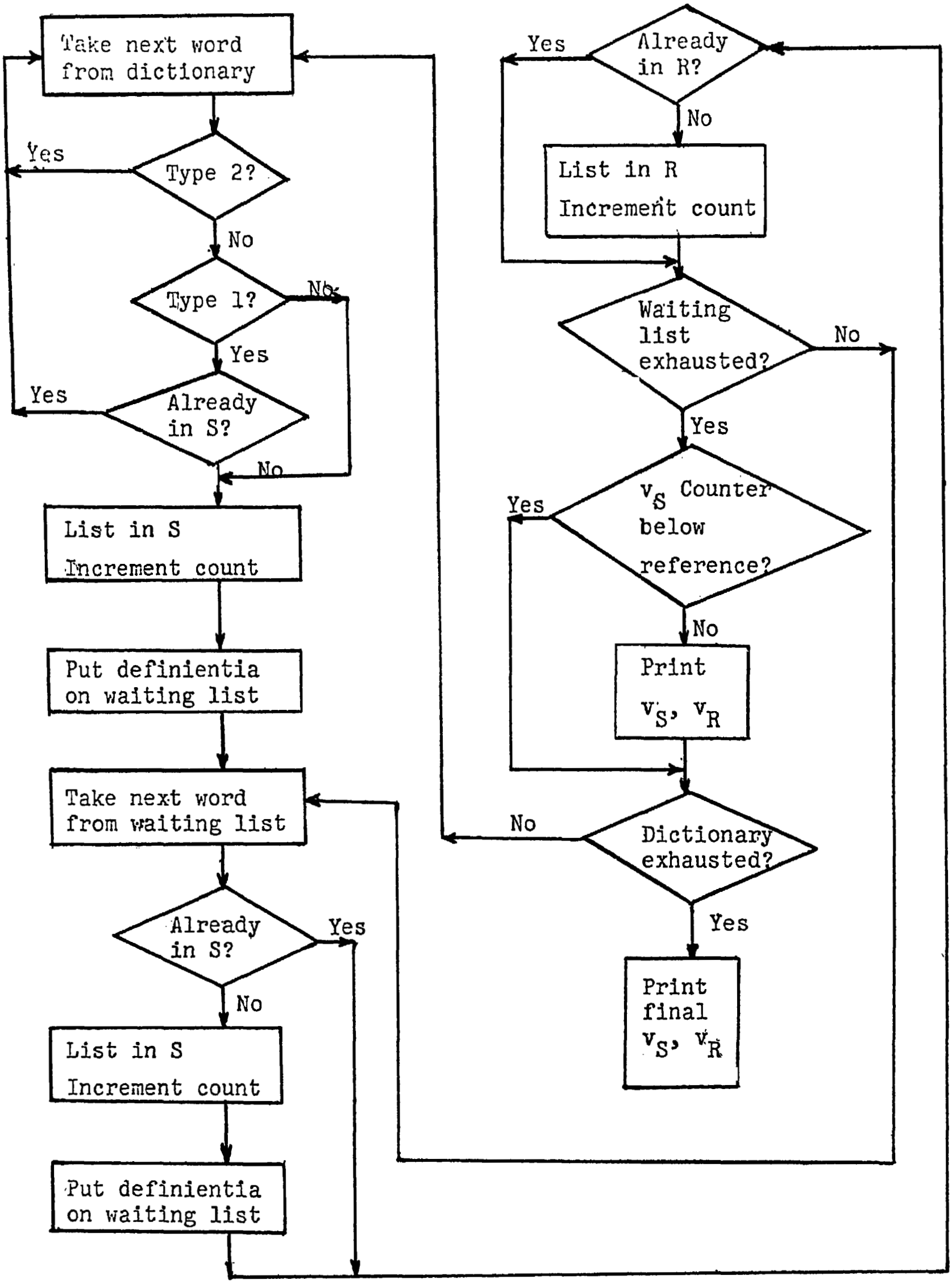


Fig. 3.—Flow diagram of COVRNG.

the full dictionary, the reference was incremented by 200; later, in the processing of the reduced dictionary, it was incremented by 100.

A criterion is needed for interrupting the program before it exceeds the time limit. An estimated increase in $\underline{v}_S$ was initially used for this purpose. A value MAXLEN was input and $\underline{v}_S$ compared with it every time a new word was added to the set. When the count reached the reference value, the program was discontinued. On the average, about 15 words per run could be added to the covered set.

Later it was found that better control could be exercised by counting the number of times that a new section of the dictionary was brought in for processing. A value MAXREP was read in and when the above counter, starting from 0, reached this value, the run was interrupted.

The variables KNTCVD and KNTCNG are counters for $\underline{v}_S$ and $\underline{v}_R$, respectively. Their current values are transferred from one run to the other. The value of KNTPRT indicates the section of the dictionary currently under investigation.

The variable INCONT is set to 0 for the very first run for each $\underline{N}$ value. This tells the routine to set up new lists for Covered List, Covering List, and a so-called Waiting List. In all successive runs its value is 1, indicating that the program

must bring these lists in from the external file.

The routine examines the current section of the dictionary, entry by entry. In the first series of runs, it deals with one of the five sections, stored in one of the five files, in the form of the original card images. A sixth file was created for storing all the lists generated by the program. When the dictionary was later reduced (for reduced values of N), the corresponding sections of the reduced dictionary were also stored in that sixth file.

If the current entry is an element of the basic vocabulary (type 2), the routine bypasses it and takes the next entry. This can be done in the processing of the full dictionary because all these words occur in the definitions and will certainly be caught later. This is no longer so in processing the reduced dictionary because the words in the definitions of which they occur may have been eliminated. In the latter case, therefore, this type of a word is immediately added to both the Covered List and the Covering List (it always covers itself).

If the current entry is a word that does not occur in any definition (type 0), it is being encountered the first time, and we are sure that it is not already on the Covered List; hence, this question need not be asked.

Otherwise the routine tests if the word is already on the

Covered List, which may well be the case because the word may have occurred earlier in the definition of another word. If so, the routine proceeds to the next word in the dictionary.

If the word is not found on the Covered List, it is put there, and KNTCVD is incremented. Then all the words in the definition of the word in question are put on the Waiting List, which is subsequently processed. This is necessary because of the adopted principle that all the covering words must themselves be covered. An entry in the v_S versus v_R table is meaningful only if this condition is satisfied.

The current dictionary entry itself is recorded as the value of the variable DREF, which passes the information on, from one run to the next, where in the dictionary the program is currently in action.

The routine then examines the Waiting List, word by word. If the current word is already on the Covered List (it may have occurred earlier in the dictionary), the routine checks if it is also on the Covering List (it may not be because it has not yet occurred in the definition of another word). If not, it is put there, and KNTCNG is incremented. All words on the Waiting List come from definitions and must therefore be added to the Covering List. After a word has been processed, it is deleted from the Waiting List.

If the current word is not on the Covered List, it must obviously be put there. First, however, the routine tests if the word occurs in the section of the dictionary currently in store by checking whether its numerical value is between those of the first and the last word of the section. If the word is not there, the routine postpones its processing and takes the next word from the Waiting List because it is more economical to process first all the words available in the dictionary section present than to read in other sections of the dictionary as the words dictate it (memory swapping is expensive).

Should the word be in that section, the routine adds it to the Covered List, increments KNTCVD, and actually looks for the word in the dictionary. If it does not find it, it gives an error message, prints out the questionable word, and terminates the run. This way the remaining punching errors in the data base were detected, and a few words were found missing (due to human error during the construction of the data base when it was forgotten to enter words that acutally occured in definitions). The files were updated accordingly.

If the word is found, the routine adds all the words in its definition to the Waiting List, then investigates its presence on the Covering List, and proceeds as described before. When the bottom of the Waiting List is reached and the list is not empty, the words remaining on it must be in other sections of the dictionary. The section present is then erased and the next

section is brought in (if the current one is section 5, section 1 is read in). The processing of the Waiting List now starts from the beginning and continues as described above.

If the Waiting List is finally empty, and KNTCVD equals or exceeds LSTREF, the routine increments LSTREF by the prescribed amount, and prints the values of KNTCVD and KNTCNG. If the count is less than the reference value, the routine simply proceeds. In any case, it tests if the proper section of the dictionary happens to be in the store (it knows that by the value of KNTPRT). If it does not, the section present is erased and the right section is read in.

Next the routine looks for the word at which it had previously stopped tracing the dictionary (it knows that by the contents of DREF). An error message has been provided for the case in which it does not find the reference for some reason. Fortunately, the program never made use of this message. After finding the reference, the routine takes the next word from the dictionary and proceeds as alread described.

When the routine reaches the bottom of the dictionary, it tests if it is the last section. If not, the next section is processed as described. At the end of the last section the routine prints the final values of v_S and v_R, and with this the processing is finished for a given value of M.

The above smooth description involves countless runs. Interruption criteria are tested at appropriate places, and the processing is discontinued accordingly. Whenever a run is terminated, the three compiled lists are saved by storing them in the external file (we shall call it File 9 for the sake of convenience). The control parameters and reference variables are printed out. The data cards are changed accordingly, for input to the next run.

The first series of runs was perfomed with the full dictionary, for which the maximum definition length N is 22. In the following series of runs N was gradually decreased. It was then also necessary to reduce the dictionary by eliminating all words with definition length greater than the current N, then eliminating all words containing them in their definitions, subsequently eliminating all words the definitions of which contain the latter, etc.

The program calls another major subroutine, named DICRED, to carry out this operation. The routine is basically simple; what makes it appear complicated is the manipulation of the files. It was found to be most convenient to search one section of the dictionary per run.

From the data cards, the routine reads a reference parameter called KNTSCT, which indicates the highest consecutive section number that has been searched. The control variable IDRP has

value 0 at input; the routine changes it to 1 if any words were removed from the section currently being searched, otherwise it remains 0 at output. The variable KNTRPT shows the number of the section currently being searched. The parameter INDFIL is set to 0 every time a new section is searched the first time. This tells the routine to bring in the section indicated by KNTSCT. If its value is 1, the section to be read is indicated by KNTRPT.

The reduced sections are stored if File 9 consecutively. If KNTRPT is less than KNTSCT, the sections following the one currently searched are stored on a temporary file because the length of the one being searched may decrease. Not until the search has ended and the current section has been stored back at its proper place are the following sections transferred back to File 9. For example, if KNTRPT = 1 and KNTSCT = 5, then sections 2, 3, 4, and 5 are stored away.

In the very first run for a given N value, i.e. if KNTSCT equals 1, the routine creates an empty list for the so-called Removal List. In the subsequent runs the routine reads in the Removal List from the file.

The routine examines the definition lengths of the entries in the current section, item by item. The entries the definition length of which is greater than the set N value are put on the Removal List and deleted from the dictionary. The value of INDP is set to 1 if such entries are found. The removed words are

printed out for reference.

Then the dictionary is searched and all definitions are checked against the items on the Removal List. If a definition containing a removed word is found, the respective entry itself is added to the Removal List and subsequently deleted from the dictionary. If a search results in any new additions to the Removal List, the search is repeated. This is continued until no new deletions occur.

After the n-th section has been processed the first time and if deletions have occurred, KNTRPT is set to 1,2, . . ., n, respectively, in n succeeding runs. If any one of these produces deletions (IDRP set to 1), the sequence is repeated. This is continued until IDRP remains in all n runs.

At the end of every run, after the temporarily saved dictionary sections have bee restored, the Removal List is stored as the last in File 9. Then the values of the key variables are printed out. The data cards are changed accordingly for the next run. After the sequence of runs with KNTSCT = 5 has been completed, the operation is finished.

The reduction was carried out with values of N equal to 16, 8, and 4. The value 10 was tried after 16, but the resulting reduction was too slight so that the series was discarded and the value 8 was used instead. At N = 4. the size ratio was already

so close to unity that a further reduction to 2 would no longer have been very informative.

All sections of all the successively reduced dictionaries have been preserved on File 9. Presently File 9 has 15 lists, each ending with an EOF. The 16-th contains the Covered List, the Covering List, and the Waiting List from the last run. These three are not separated by EOF's as there was no necessity for separating them. This list collection has no particular importance.

The remaining subroutines in the program are short auxiliary routines for aiding the principal routines where needed. The function INPUTL reads in a list structure from the card images on file, without printing out the list as does the original SLIP routine. It constructs erasable local sublists. It is virtually the same routine as READLS used by ANALEX.

RESTOR is equivalent to the SLIP subroutine of the same name except that it does not leave a SLIP cell with a list name as datum floating in the available space. (The latter tends to cause program termination with an error message to the effect that a list was required but not found.)

The subroutine SKIP is needed for convenient accessing of the various lists in File 9. Finally, the function DLTLST is the most effective means so far tried for deleting list structures

built by the SLIP routine BNINPL. (It does not completely destroy them, however, and if BNINPL is used repeatedly, the store is still gradually filled with residues that make available space unavailable.)

APPENDIX II

Some Ideas for the Program to Investigate the Relationship Covering Set Size versus Maximum Definition Length

The second proposed problem, viz. finding v_R as a function of N for fixed values of v_S, is diccussed now. This will be a task of proportions no less than the present, except for the construction of the data base. The following procedure, represented by a simplified flow chart in Figure 4, is suggested for carrying out this task.

INSERT FIGURE 4 ABOUT HERE

The program starts with known values of N and v_R (in this case 22 and 1,464, respectively). It first replaces words in R having a definition length of 1 (except, of course, those defined by themselves) by their definition in all definitions. Then the program looks for words of short definition length in R (x = 2,3,4, etc.). It substitutes their definition for them in all definitions and counts them out from v_R. Simultaneously, it keeps track of possible increase in N due to this process and

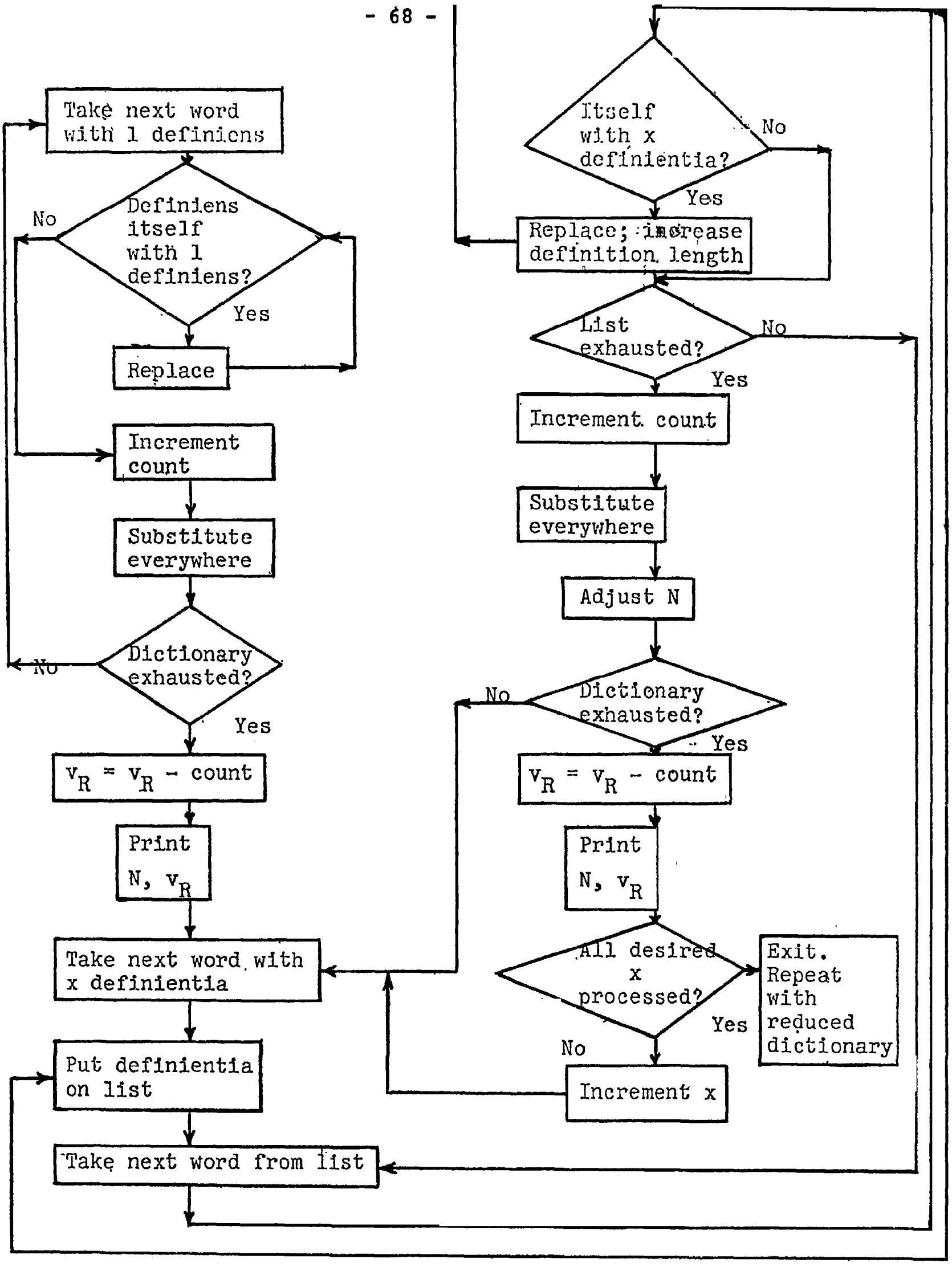


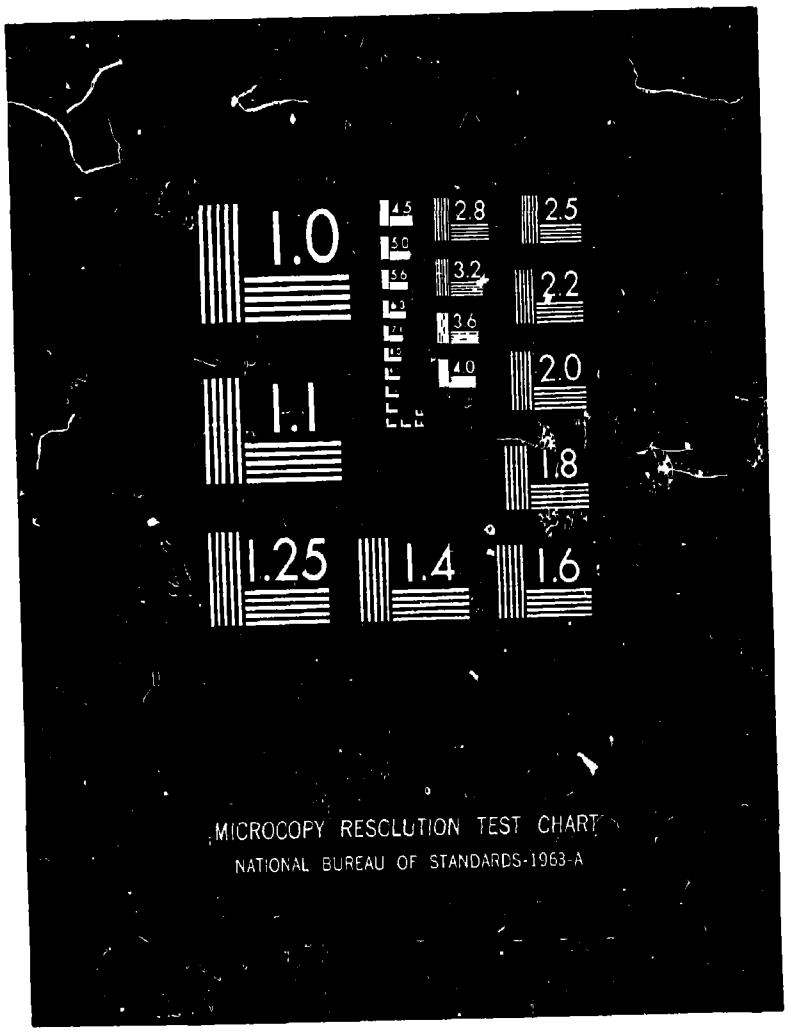
Fig. 4 .—Flow diagram for establishing N-v_R relations

records the value. The process is repeated with reduced dictionaries, which have different $\underline{vs}$ values.

As pointed out earlier it is not suggested that definitions so created are usable or acceptable to the speaker of a natural language. The procedure, however, will produce the numerical relationships desired.

The existing data base, together with its reduced versions, has been stored on magnetic tape and is ready to be used as input into the proposed procedure.

END


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A