6

1965 International Conference on Computational Linguistics

AUTOMATIC LINGUISTIC CLASSIFICATION

E. D. Pendergraft and N. Dale

Linguistics Research Center
The University of Texas
Austin, Texas, U. S. A.

ABSTRACT

Many investigators have recognized that sophisticated mechanical translation or other computational linguistic systems will require language learning capabilities, both in the ability to discover and verify new descriptive rules, and to adapt rules to new situations of use.

The work plan of a long-range series of experiments in automatic linguistic classification is described, together with a discussion of the first experiment now in progress. The latter is concerned with category identification. In particular the data resulting from automatic syntactic analysis of English texts will be used to identify syntactic categories which have similar membership.

The series of experiments will accordingly combine the use of automatic linguistic analysis and automatic classification techniques. Automatic syntactic analysis, and in later experiments semantic analysis, will be performed within the Linguistics Research System (LRS). Automatic classification will be carried out within the Automatic Classification System (ACS). Both of these computer systems have been developed by the Linguistics Research Center.

LRS is a large-capacity system designed especially to support research in computational linguistics. It currently has the capability of performing several types of generalized translation, and of transferring information among any number of languages through the use of interlingual descriptions. This system operates under its own monitor.

ACS is a Fortran IV system operating under the IBSYS monitor. It performs a variety of classification operations on large universes of objects having specified properties. Either objects or properties may be classified, and correlations may be computed among resulting classes.

A programming interface is being constructed between these two systems so that their combined capabilities can be used for automatic linguistic classification.

1 INTRODUCTION

Mechanical translation research over almost two decades has led to a broader discipline, computational linguistics, which already includes within its concern the automated processes that collect, store, retrieve or communicate information conveyed in or about language, as well as translate one language into another. With progress in automatic classification, another possibility is being explored, that of creating new information rather than merely gathering, maintaining or distributing the results of human intellectual activities.

Many investigators have noted that sophisticated linguistic systems must be capable of learning. A new term may be defined within a text being translated mechanically. Or, more commonly, a new meaning may be given a term used in close communication among colleagues. Bar-Hillel, for one, has waxed and waned in enthusiasm for mechanized linguistic learning, finally relegating even its investigators to the "lunatic fringe" of computational linguistics. Some thoughtful work has been done by Lamb [1] . Certainly Solomonoff [2] should be mentioned, as should Knowlton [3] and Sparck-Jones [4], but each for a different reason. There is hardly a literature to cite, unless it be that unruly assemblage we have come to call "artificial intelligence."

The name "self-organizing system" has also come into use. We will adopt it, so that "learning" and "adapting" may distinguish different kinds of self-organization.

We observe, furthermore, that to process information one must first process the language (or symbolic system) in which the information is given. As a consequence, every information processing system has a component that processes linguistic information. And that component, we now know, may have a subcomponent which processes meta-linguistic information.

Self-organizing linguistic systems properly fall within the scope of meta-linguistic processing. The information being processed is about some language, the "object-language" of the communication; hence the vehicle by which the information is conveyed is a "meta-language." If the self-organizing system has changed the description of the conventional alternatives available within the object-language, then we will say that the system is "learning." Whether or not the alternatives remain unchanged, if some alteration has been made in the conventions of their use, the system will be "adapting."

Thus, roughly speaking, learning will involve some change in linguistic rules that describe a set of well-defined alternatives in the object-language. Adaptation will involve some change in a set of probabilities that describe how those alternatives are being used.

A self-organizing information system, in contrast to one learning or adapting by meta-linguistic processing, would employ linguistic processing to create new information about some subject-matter not necessarily linguistic. But since the information so processed might indeed be about language, we anticipate that linguistic self-organization may be based either meta-linguistically or linguistically.

For the present, however, our system will be based on meta-linguistic processing. Work in meta-syntactics is progressing rapidly; researchers in computational linguistics now face the obligation of testing hypotheses more rigorously than with heuristic arguments or typical linguistic examples. More careful investigation is need in meta-semantics, i.e. in the relations between meta-linguistic and linguistic information.

In essence, then, we will try with automatic linguistic classification to bridge the gap between the design of language and the events of spoken or written discourse. What we have to report is only a small beginning toward that objective.

We recognize that these are difficult problems requiring long-range commitments. They are nevertheless central to improving the language data used in automated analysis, synthesis and translation. Moreover, they lead to the concept of a dynamic language data base in linguistic processing.

Already it is clear that the amount of information contained in a language description greatly influences efficiency in linguistic processing. Contrary to our former intuition, a simple description may merely be deficient in information so that the search in automated analysis will be extended unduly. There appears, furthermore, to be an optimal size in the syntactical descriptive unit. Thus, in making the transition from syntactical to semantical description (at least for the theories [5] we are studying), the basic question is analogous to that in the transition from lexical to syntactical description which gives rise to morphology: viz. what objects are to be classified? We are attacking these semological and morphological problems within the same theoretical structure that determines how the resulting objects are to be classified. Indeed, the two questions appear inseparable.

2 BASIC PROGRAMS

The Automatic Classification System (ACS), a Fortran IV programming package (7) based on the classification theories of Needham and Parker-Rhodes (8) has been developed by the Linguistics Research Center of The University of Texas (under support of the National Science Foundation and the U. S. Army Electronics Laboratories), and has been made generally available for classification research. The version of the system used in our own facility has been augmented with list-processing routines and other specialized programming which greatly increased its efficiency and data-capacity.

ACS is a generalized classification system which can be applied to non-linguistic as well as linguistic problems. Its basic inputs are data describing the incidence (or the frequency of incidence) of particular properties upon particular objects. These incidence data may be transposed, so that either the properties or the objects can be classified. Various measures of the similarity between pairs of the objects (or properties) are available, permitting the incidence data to be used in computations of the connections between object (or property) pairs. Using these connection data, other routines group together "clumps" of objects with similar properties (or of properties occurring similarly in the objects). Various kinds of the clumps can be discovered. ACS has a section which controls the selection of similarity measures and clumping methods in classification experiments (6).

To formalize the concepts of distributional classification, e.g. those investigated by Hockett (8) and by Harris (9) we have extended the general classification theory to binary as well as singulary relations. In linguistic classification these can be interpreted as constitutive relations (e.g.

concatenation). This interpretation, more exactly, assumes that the incidence data describe pairs of objects standing in that particular relation. Clumps of similar objects are then found in both the domain and counterdomain of the relation. Finally, individual clumps in the domain are paired with individual clumps in the counterdomain according as the connections between members of the two clumps are dense (in a precise sense) relative to the entire set of connections. Pairs of clumps may also be found by using a measure of relative sparseness in the connections. These capabilities have been added to ACS, and programming is being done to prepare incidence data mechanically from the results of automatic analysis in LRS (10).

The automatic analysis algorithms in LRS are linguistically generalized, i.e. they will recognize the expressions of any object-language according to the exact specifications described in particular meta-languages. These language data, furthermore, are operationally generalized; they will be given relationally (solely in terms of relations and not as a process) so that synthesis as well as analysis algorithms may refer to the same descriptions. Object-language descriptions are conveyed by a hierarchy of meta-languages rather than by a single meta-language (5). The complete data hierarchy will be given by lexical, syntactical, semantical and pragmatical meta-languages; the first three are currently available in LRS. Lexical, syntactic, semantic and pragmatic analysis (or synthesis) algorithms will be oriented to the corresponding levels of monolingual data. Analysis will affect a transfer to the next higher level of processing; synthesis to the next lower level. Automated lexical and syntactic analysis (as well as synthesis and translation) are operational in LRS, and the semantic algorithms will be later this year. All of the algorithms are parallel, stochastic, heuristic and machine-independent; that is to say, they have the following design features which we believe to be important in automatic linguistic classification experiments:

(a) They carry forward a search for all possible linguistic alternatives in parallel, instead of following to completion one sequence of alternatives before beginning another. As a result, all of the available linguistic evidence is represented in the analysis output.

(b) They compute a probability for each linguistic alternative being processed. The probability will be the same in analysis or synthesis; it represents the likelihood of occurrence in the language rather than in the process.

(c) They (may or may not, as a matter of choice) use the probabilities as heuristic criteria to limit the analysis output to the most likely alternatives.

(d) They are oriented entirely to particular meta-syntactical and meta-semantical relations, not to the components of a particular computer. All processing decisions and results will be the same on every computer large enough to do the linguistic processing.

3 MORPHOLOGICAL CLASSIFICATION

Constitutive relations may, of course, be used as the basis of classification of individual objects (i.e. for singulary as opposed to binary classification). A formal distinction can also be made between the classes of objects which are identical and those which are to some degree equivalent in distribution. The former, to be specific, have identical clump membership and are thus indistinguishable with respect to the particular constitutive relation described in the incidence data. The latter have common membership in a particular set of clumps and, as a consequence, share certain distributional properties which are represented by those clumps.

Morphological classification, therefore, will involve the following basic operations within our theories. (These will also be pertinent to our remarks below about semological classification.) For the given constitutive relation, the classification algorithm will have to:

(a) Recognize, among the objects potentially made available by segmentation, those which are to be classified.

(b) Perform singulary classification of the recognized objects to determine which subsets of them have identical distribution relative to that constitutive relation.

We assume, in the morphological problem, that the objects to be classified are lexical units (whether phonetic or orthographic) and that concatenation is the constitutive relation. Our working hypothesis is that the morphological objects are those which maximize the connection entropy. This seems intuitively reasonable, since more or less homogeneous connections would be anticipated among objects having elementary status. Conversely, a relatively strong connection between two objects would be evidence that they were parts of a single construct.

Accordingly, a routine has been added to ACS to normalize the connection data and, for the normalized connections $p_1, p_2, \ldots, p_2$ $(1 \geq p_i; \sum_{i=1}^{n} p_i = 1)$, to compute the connection entropy

$$H(p_1, p_2, \ldots, p_n) = -\sum_{j=1}^{n} p_j \log p_j$$

for a convenient logarithmic base [11]. The second operation, that of determining what objects have identical distributional properties, will be handled by a routine which finds the sets in the intersections defined by the collection of all clumps. (We will say that the members of the identity classes form "component sets" of the universe of objects being classified, because the sets partition that universe.)

Some of the strategies which may be used in morphological classification have been compared by Hockett [12] who concluded that different classification methods could succeed in establishing the same relation between morphemes and phonemes. A strategy chosen for automation must, above all, be computationally tractable. The two methods which Hockett calls the "morph approach" and the "morphophonemic approach" would have inherent advantages or disadvantages computationally.

The "morph approach," according to Hockett, supposes that morphemes are *represented* by morphs and that morphs are *composed* by phonemes. In consequence, the constitutive relation (concatenation) must obtain between constructions of phonemes. By our hypothesis, a set of morphs would be any set of the constructions maximizing (perhaps locally) the connection entropy. And members of each identity class of the morphs would be the allomorphs of a particular morpheme.

Computationally, then, we might carry out the following morphological classification algorithm:

(a) Perform lexical analysis, using the current set of phoneme constructs as the lexical data (lexicon).

(b) Prepare incidence data describing the constitutive relation between the contructs.

(c) Compute the connections between pairs of the constructs and the connection entropy, comparing the result with the entropy of the preceding cycle.

(d) If the entropy has increased, combine the (one or more, depending upon the rate of increase) strongly connected pairs into single constructs; return to (a).

(e) If the entropy has not increased, perform singulary classification and find the component sets of morphs which will represent morphemes.

Hockett's "morphophonemic approach," in contrast, would take morphemes to be <u>composed</u> of morphophonemes and morphophonemes to be <u>represented</u> by phonemes. One interpretation of these relations in terms of the classification theories (among several) would involve the supposition that the members of each identity class of the phonemes represent a particular morphophoneme. Consequently, any phoneme representing the morphophoneme M would be distinguished in the incidence data only as an M. A phoneme construct, similarly, would be distinguished only as the construct of represented morphophonemes. Any set of the morphophoneme constructs maximizing (maybe locally) the connection entropy would be recognized as morphemes.

That these relations would call for a different computational strategy should be evident. Because of the higher level of abstraction, one might anticipate that (a) there would be fewer morphophoneme than phoneme constructs, and (b) the latter would occur more frequently than the former in the outputs of lexical analysis. But our aim is not to prejudge the computational advantages of one approach above another; the schemes which are feasible within our analysis and classification capabilities will be tested.

4 SYNTACTICAL CLASSIFICATION

The advantages of abstraction in classification are nevertheless striking in syntactical applications. Indeed the possible gains seem so promising that we have bypassed automated morphological classification in our first experiments to investigate the following operations of syntactical classification. Each operation presupposes not only the existence of a set of morphemes, but an assignment of the morphemes to syntactical equivalence classes relative to concatenation, as already described.

(1) Identification of classes. If, in the outputs of syntactical analysis, it is found that some expression has been (ambiguously) recognized both as an A and as a B, then this coincidence of A and B will be the event counted. Singulary classification will then be performed to determine whether an A and a B are distinguishable distributionally relative to coincidence. If not, we will induce that the predicates "A" and "B" are coextensive, i.e. they denote the same objects [13]. The two predicates will therefore be replaced (wherever they occur in the syntactical description) by a single predicate.

(2) Generalization of classes. During the class identification operation (1), the event of being an A will be assigned to a set of (zero or more) clumps. If being an A entails being in the clump C, then we introduce the new predicate "C". We induce, further, that the predicate "C" comprehends the predicate "A", i.e. "C" denotes every object that "A" does [13]. And, since the extensions of the new predicates are clumps of objects sharing some distributional property, we characterize "A" and "C" as ostensive and distributional predicates, respectively, relative to the constitutive relation.

Taking a new incidence data to describe the relation of comprehension between the distributional and ostensive predicates, we will next perform singulary classification to bring together predicates which are similar relative to comprehension. Thus, we induce that the predicates have similar extensions. The ostensive predicates in each clump will be replaced (in all their occurrences in the syntactical description) by the distributional predicate of that clump, i.e. by the predicate whose extension is the union of the extensions of the extensionally-similar predicates. (K-clumping is convenient for generalizing classes because it provides a parameter for the degree of generalization.)

(3) *Rule generation*. The aim of this operation will be to find new syntactic rules (i.e. taxonomic axioms) to be added to the syntactical description. The events to be counted in preparing the incidence data will be those in which an A is found to be concatenated to a B in the outputs of automated syntactic analysis. Binary classification will be used to pair clumps on the basis of dense connections, as explained above. For any resulting pair of densely connected clumps C, D classifying an A as a C, and a B as a D, respectively, we generate the syntactic rules $A \subseteq C$, $B \subseteq D$ and $C^\frown D \subseteq E$. The predicate "$C^\frown D$" will have as its extension any C concatenated to any D. "E" will be a new predicate comprehending "$C^\frown D$."

Rules generated inductively will tend to be overly general. There will be an operation, however, by which syntactical classes can be specialized to conform to the empirical analysis data.

(4) *Specialization of classes*. From the rules $A \subseteq C$ and $C^\frown D \subseteq E$ we may infer the derived rule $A^\frown D \subseteq E$. Hence the *application* of $A \subseteq C$ to $C^\frown D \subseteq E$ at the first (left-most) place

in the latter may be symbolized algebraically [14] as follows:

$$(C \frown D \subseteq E) \frown (A \subseteq C) = (A \frown D \subseteq E)$$

Incidence data, prepared for one particular class C, will describe the frequency of application of rules at places mentioning that class. The events counted, specifically, will be those in which a rule X is found (in the analysis outputs) to be applied at a place p in rule Y (i.e. for the event $X \overset{\frown}{p} Y$, the pair of objects Y^p,X will be regarded as standing in the constitutive relation). Different places in the same rule will be treated as different objects relative to application. Binary classification will be used to pair densely-connected clumps of distributionally similar (in the domain) places of application, and (in the counterdomain) rules being applied to the places. The predicate "C" will be replaced (in those particular occurrences in the syntactical description) by a new predicate denoting that subclass of C.

These syntactic classification operations will be operational in the combined LRS-ACS programming system before the end of this year. We plan an extension of the system to include automated morphological, semological and semantical classification. The last will be restricted to a distributional semantics without identification of references, i.e. to the restricted form of our theoretical hypothesis [5] which assumes that applications at different places in the same rule are independent events.

5 SEMOLOGICAL CLASSIFICATION

Recently we observed [15] that a small informational unit in language data seems convenient for the descriptive linguist, but a large informational unit would optimize linguistic processing. In retrospect it appears likely that, in our project and elsewhere, different approaches to syntactical description have too often been concerned with different informational units rather than different information. As anticipated above, we have come to questions in syntactical classification which are analogous to those in lexical classification which gave rise to morphology; viz. *what* objects are to be classified semantically?

Joos [16] has stimulated our thinking about semology, as has Lamb [12]. Undoubtedly the latter's own interest in automated syntactical classification [1] has contributed to the similarity of our theories; the study of automatic linguistic classification brings one to consider informational units which are small enough to be discovered mechanically.

Adopting Bloomfield's terminology [18], we will refer to the elemental units of syntactical description (i.e. those rules conveying minimal units of information) as *tagmemes*. The elementary units to be classified semantically will be semes. Between the two, we will posit semological relations analogous to those which Hockett presented for morphology.

(1) The first hypothesis would be that sememes are *represented* by semes, and that semes are *composed* of tagmemes. Within the frame of our classification theories, therefore, the constitutive relation would be *application*: the semes would be regarded as the representatives of a particular sememe. This is the approach we will take in our first semological experiments.

(2) Semes would be <u>composed</u> of semotagmemes, in the second hypothesis, and semotagmemes <u>represented</u> by tagmemes. Consequently, for the purposes of automated classification, the members of an identity class relative to application would be regarded as the representatives of a particular semotagmeme. A set of semotagmeme constructs (locally) maximizing the connection entropy would be recognized as semes. This approach to automated semological classification may have the advantage of a higher level of abstraction, like the analogous morphophoneme approach in morphological classification.

Both semological hypotheses will be tested when we have the additional data-capacity which a magnetic disk will provide in ACS early next year. LRS programs that maintain either type of semological data are already operational.

6 SEMANTICAL CLASSIFICATION

Sememes, in the sense which may be formalized as suggested above, are regarded in our working hypothesis as describing signs in the object-language. To be specific they will have two epistemological functions:

(a) They will convey the (formational) syntax of the object-language, i.e. the information needed to construct complex signs from the basic ones.

(b) They will be units substituted in translation, paraphrasing and other transformations based on semantical criteria.

A fundamental principle leading to distributional semantics was cited by Martin [13] in 1958. In discussing "translational" and "non-translational" semantical meta-languages, he presents a thesis which we will paraphrase very roughly for our present purpose:

Semantical relations (e.g. denotation, designation), in requiring as their arguments both signs and their objects (denotata, designata), make it necessary that the semantical meta-language itself have signs for the same objects as the object-language. The meta-language signs are, accordingly, translations of the object-language signs, since the two sets have common objects. As a consequence of this, semantical relations in the meta-language will be at least as complex as those in the object-language. However a "non-translational" semantical meta-language may describe a relation between signs, but one defined in semantical terms (e.g. comprehension, where one sign will comprehend another if the former denotes every object the latter does). This second type of meta-language will be semantically less complex than the object-language.

Furthermore, as we have suggested above, it is probable that comprehension of signs may be induced from distributional evidence. A distributional semantics, in addition to being a non-translational in Martin's sense, would define comprehension or some alternative relation between signs in purely distributional terms, leaving aside all theoretical references to objects which the signs may or may not have. This is the approach we have taken, by employing the concepts of classification theory to formalize those of distribution.

With few exceptions the computational strategies in semantical classification will be the same as in the distributional syntactics. Analogous operations of class identification, generalization and specialization will be available. But the members of syntactical classes will be syntactic rules. And the rules in a given class will be required to have the same "degree," i.e. the same number of those predicates with the equivalence (but not the identity) classes as their extensions [5].

Generation of semantic rules will likewise be analogous to the syntactical operation. But our semantical hypothesis requires that all of the syntactic rules in the extensions of two semantical classes be applied (pairwise) at places of application with the same name. For instance $A \overset{\frown}{2} B$ would describe the applications of the rules in semantical class B to those in the class A at the places named by the numeral 2. When the syntactic rules are first generated, their places of application will be named positionally (from left to right) and, in the restricted theory, uniquely (no two places will have the same name). Binary semantical classification, as part of the rule generation operation, will show how the places should be renamed to satisfy the above semantical convention. (In LRS this is the information conveyed by "superscripts" associated with the appropriate predicates in syntactic rules.) Otherwise the generated semantic rules will be formally the same as the syntactic (e.g. $A \subseteq C$, $B \subseteq D$, $C \overset{\frown}{\scriptstyle P} D \subseteq E$). The numeral naming the place of application of two semantical classes is given in our notations as part of the connective symbolizing application. Conventions for renaming the places during deductive inference have been reported elsewhere [19].

7 SELF-ORGANIZING LINGUISTIC SYSTEMS

Automatic linguistic classification will give us various capabilities for changing language descriptions. We plan to study each capability separately so that it will receive its own development. Coordination of the capabilities into an integrated system will be approached as a different problem, that of self-organization. The system as a whole must not only change, but change for the better.

Homeostasis, as explained by Ashby [20], is the fundamental control principle we will investigate. Roughly speaking, it calls for reorganization when the situation (according to some criterion) is getting worse and stability when it is getting better. Hence the algorithms we described for morphological (or semological) classification were too simple. If a decrease in connection entropy defines "getting worse" in morphological (or semological) classification, the system must be able to deliver smaller as well as larger constructs during its reorganization. In syntactical (or semantical) classification, stability or reorganization (in response to decreasing or increasing entropy, respectively) may be obtained by a choice between the class identification and generalization operations. With K-clumping, class generalization may also be parameterized to specify a greater or lesser reorganization in descriptive categories.

These basic control techniques will be tried toward the end of this year. To control class specialization and rule generation, we will use the following processing sequence after each cycle of syntactic (or semantic) analysis.

(a) Compute the connections and connection entropy for each class.

(b) Sort the classes so that those with the lowest entropy come first.

(c) Perform the class specialization operation on the successive classes until one is reached which cannot be specialized.

(d) Use only that class and the ones following it for rule generation.

Underlying this processing strategy is the assumption that stable classes will be characterized by high connection entropy. (Though plausible, this must be tested.) Rule generation will thus be limited, as a result of the strategy, to those classes which are found to be the most stable. Broadly effective control strategies are our present concern; we believe it will be possible to supplement these with more selective controls later on.

Incidence data for our first automatic linguistic classification experiments have been prepared mechanically from statistics brought directly to ACS from the analysis outputs in LRS. For the self-organizing linguistic system we felt that the statistics should be accumulated from analysis statistics from LRS to the Information Maintenance System (IMS), a coordinate information storage and retrieval system [21] which we have programmed for the Aeronautical Systems Division, Air Force Systems Command. This system has been released by its sponsor for use in linguistic research. Classification statistics from ACS will also be stored in IMS. A report generator will be added to IMS so that the analysis and classification statistics can be displayed in formats suitable for publication.

Programming to implement the Self-organizing Linguistic System (SLS) will include the following routines:

7.1 LRS-IMS Interface

Transportation of the analysis statistics on *coincidence*, *concatenation* and *application* at the different linguistic levels will be performed by these programs. In addition to collecting and organizing the statistics, they will update the stores in IMS, also handling the additions and deletions of rules or classes. Normalizing factors will be maintained cumulatively so that statistics collected during different periods of time may be compared. These programs are now almost completed.

7.2 IMS-ACS Interface

This set of programs will carry out the control strategies we have mentioned. They are being written under IBSYS so that they will be compatible with ACS Programming. The IMS store has been designed so that it can be manipulated under either the LRS operating system or IBSYS. It is anticipated that most of these routines will be in operation before the end of 1965.

7.3 ACS-IMS Interface

Classification results will be collected, organized and transported to IMS by these routines. They will also update the IMS store. Their completion will coincide with routines in the IMS-ACS interface.

7.4 IMS-LRS Interface

The same request formats which the linguist uses in adding, changing or deleting language data in LRS will be used by the self-organizing system. However, language data processing in LRS may be performed either with mnemonic symbols or numerals as the names of syntactical (or semantical) classes. The automated system will use the numerals, referencing its requests to the results of automatic classification.

Because the self-organizing system will be able to make extensive changes in the data base, which would be prohibitive by manual coding, we plan to provide macro-requests (e.g. a request to eliminate the distinction between the predicates grouped together by the generalization operation).

8 AN EXPERIMENT

This experiment in class identification will exemplify the type of research we are performing. Although the operations performed are those described above as class generalization, by setting the K-clumping parameter to 1 we obtain component sets as the classification output.

8.1 Experimental Design

General Definitions

Given a binary matrix:

1(i,j): the number of 1's in the intersection of columns i and j.

1(i): the number of 1's in column i.

1(j): the number of 1's in column j.

Phase 1: Forming Connection Matrix

Part 1: Construct Incidence Array A

$A = (a_{i,j})$ such that

$a_{i,j} = 1$ the j^{th} object is described by the i^{th} property.

$a_{i,j} = 0$ 0 otherwise

Part 2: Compute Frequency Matrix from A

$F = (f_{i,j})$ such that

$f_{i,j} = 1(i,j) \quad i \neq j$

$f_{i,i} = 1(i)$

$f_{j,j} = 1(j)$

Part 3: Reduce Dimensions of F

Remove rows and columns of F as follows:

row (column) i is deleted

$\Longleftrightarrow f_{i,j} = 0$ for all j, $j \neq i$

Part 4: Normalize columns of F

$N = (n_{i,j})$ such that

$(N_i)_j = f_{i,j}/f_{j,j}$

Part 5: Compute Connection Matrix C

$C = (c_{i,j})$ such that

$c_{i,j} = \sum_{k=1}^{m} \min (n_{i,k}, n_{j,k})$ where m is the number of columns (rows) in the normalized reduced matrix F.

Discussion: The frequency matrix which forms the incidence data for the experiment is a table showing how many times an object i coincides with an object j. Reducing the matrix by removing columns which are all zero on all off-diagonal cells, deletes from the set of objects those objects for which there are no coincidence data. Normalizing the columns of F by the diagonal--which contains the number of instances of the object in the sample--produces the normalized incidence matrix.

Connection matrix C is a symmetric matrix which describes the relation of object i to object j based on the normalized incidence data. Matrix C constitutes the data for the next stage of processing.

Phase 2: Locating GR-Clumps

Definitions

C: Connection Matrix computed in Phase 1.
U: Universe set (set of objects characterized in connection matrix C.)
A: A subset of U.
$\bar{A}$: $U-A=\bar{A}$, complement of A.
x: An element of U.
a_i: An element of A $(a_1, a_2, ..a_t)$
$\bar{a}_i$: An element of $\bar{A}$ $(\bar{a}_1, \bar{a}_2, ..\bar{a}_r)$
(Note: $r+t=m$)
$c(a_i, a_j)$: $c_{i,j}$ connection of object i to object j as defined in the i j^{th} cell of C.
$C(x,A)$: $\sum_{i=1}^{t} c(x, a_i)$
$C(x,\bar{A})$: $\sum_{i=1}^{r} c(x, \bar{a}_i)$
$b(x,A)$: $C(x,A) - C(x,\bar{A})$ The bias of objects x to the set A is the excess (either positive or negative) of its connections to A less its connections to A.

$$A \times A: \sum_{i=1}^{t} \sum_{j=1}^{t} c(a_i, a_j)$$

$$\bar{A} \times \bar{A}: \sum_{i=1}^{r} \sum_{j=1}^{r} c(\bar{a}_i, \bar{a}_j)$$

$$A \times \bar{A}: \sum_{i=1}^{t} \sum_{j=1}^{r} c(a_i, \bar{a}_j)$$

GR-clump: U set A is a GR-clump of U $\Longleftrightarrow$ it is a local minimum for the following function

$$F(A) = \frac{A \times \bar{A}}{A \times A + \bar{A} \times \bar{A}}$$

In terms of individual elements, the definition can be stated as follows:

$$A = \{x \mid b(x,A) \geq 0 \forall x \epsilon A \text{ and } b(y,A) < 0 \forall y \epsilon \bar{A}\}$$

Discussion: There is no known way to predict how many GR-clumps exist in a given space. The GR-clump finding procedures [22] produce a set of highly overlapping GR-clumps.

Phase 3: Forming K-Clumps of Objects

Part 1: Form an object-GR-clump Incidence Array A

$A = (a_{i,j})$ such that

$a_{i,j} = 1$ if object j is in the ith GR-clump.

$a_{i,j} = 0$ otherwise

Part 2: Form a Connection Matrix F

$$F = (f_{i,j}) \text{ such that}$$

$$f_{i,j} = \frac{l(i,j)}{l(i)+l(j)-l(i,j)}$$

$$f_{i,i} = f_{j,j} = 0$$

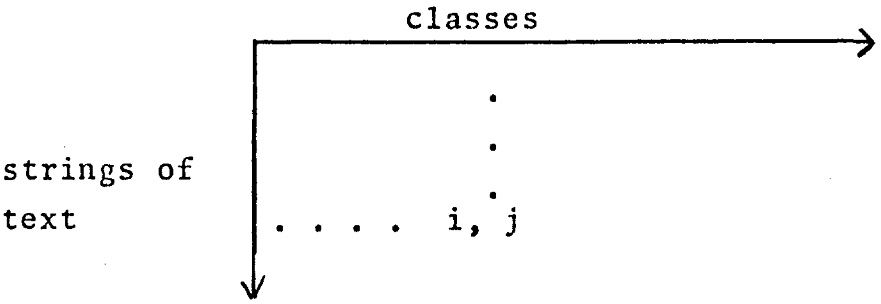
Part 3: Locate K-clumps in F

Discussion: The K-clumps located in F will be those elements which are highly similar in their distributional properties. The threshold value can be used to vary the amount of similarity.

8.2 The Experiment

Phase 1:

Data Base: Six paragraphs of English text were syntactically analyzed in LRS. The outputs were on magnetic tape. A computer program was written to take this data and form a binary incidence array as follows:



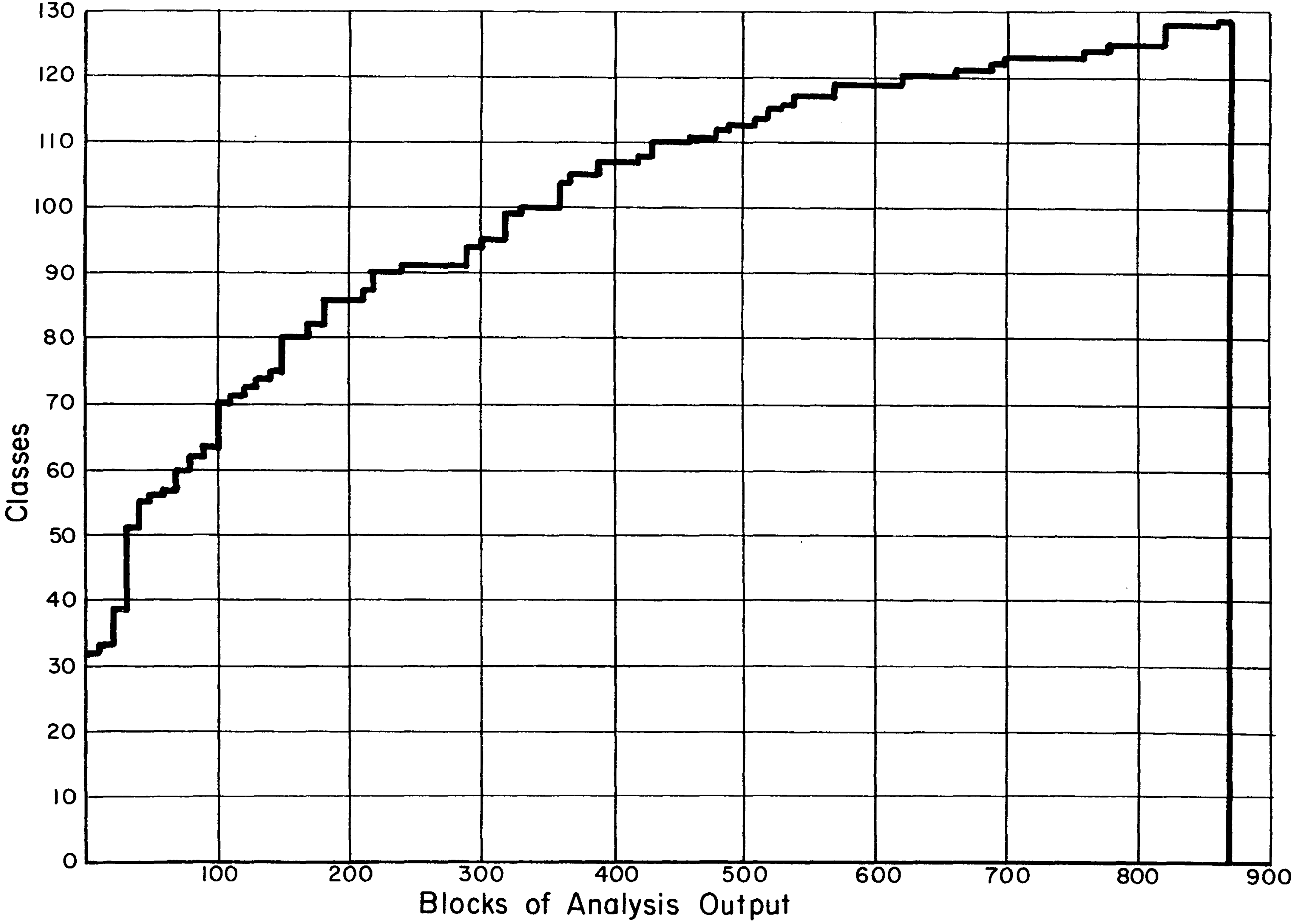
i, j = 1 if string i was in class j

The list of classes was generated at the same time. In the six paragraphs, 129 classes were found. Graph 1 shows the rate at which classes were found.

In the next stage of processing, this incidence array was used to make a co-incidence frequency count. Forty-five of the 129 classes occurred uniquely, i.e. did not coincide with another class. These 45 were deleted from the data set, leaving 84 classes.

The next step was to normalize the frequency matrix and compute the connection matrix as explained in the experimental design. In the 84 x 84 matrix there were 1012 nonzero entries giving a matrix density of 14.3%. The connection values ranged from zero to 3.33283.

GRAPH 1: BUILD-UP OF CLASSES



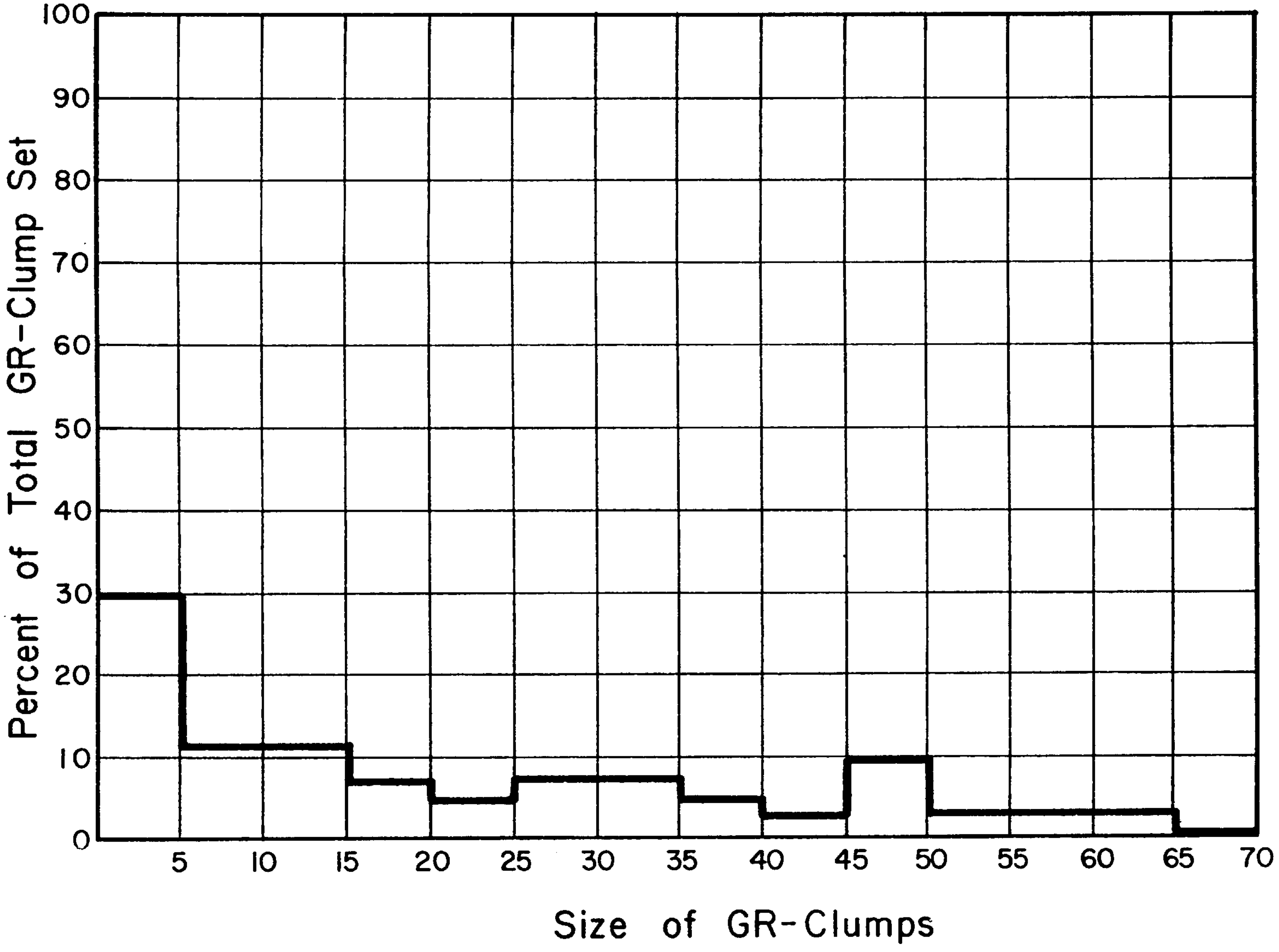
Phase 2:

GR-clumping was done in the connection matrix describe in Phase 1. Using the pivot variable method of initial partitioning [22] 44 GR-clumps were located. Graph 2 displays the distribution (by size) of the GR-clumps found.

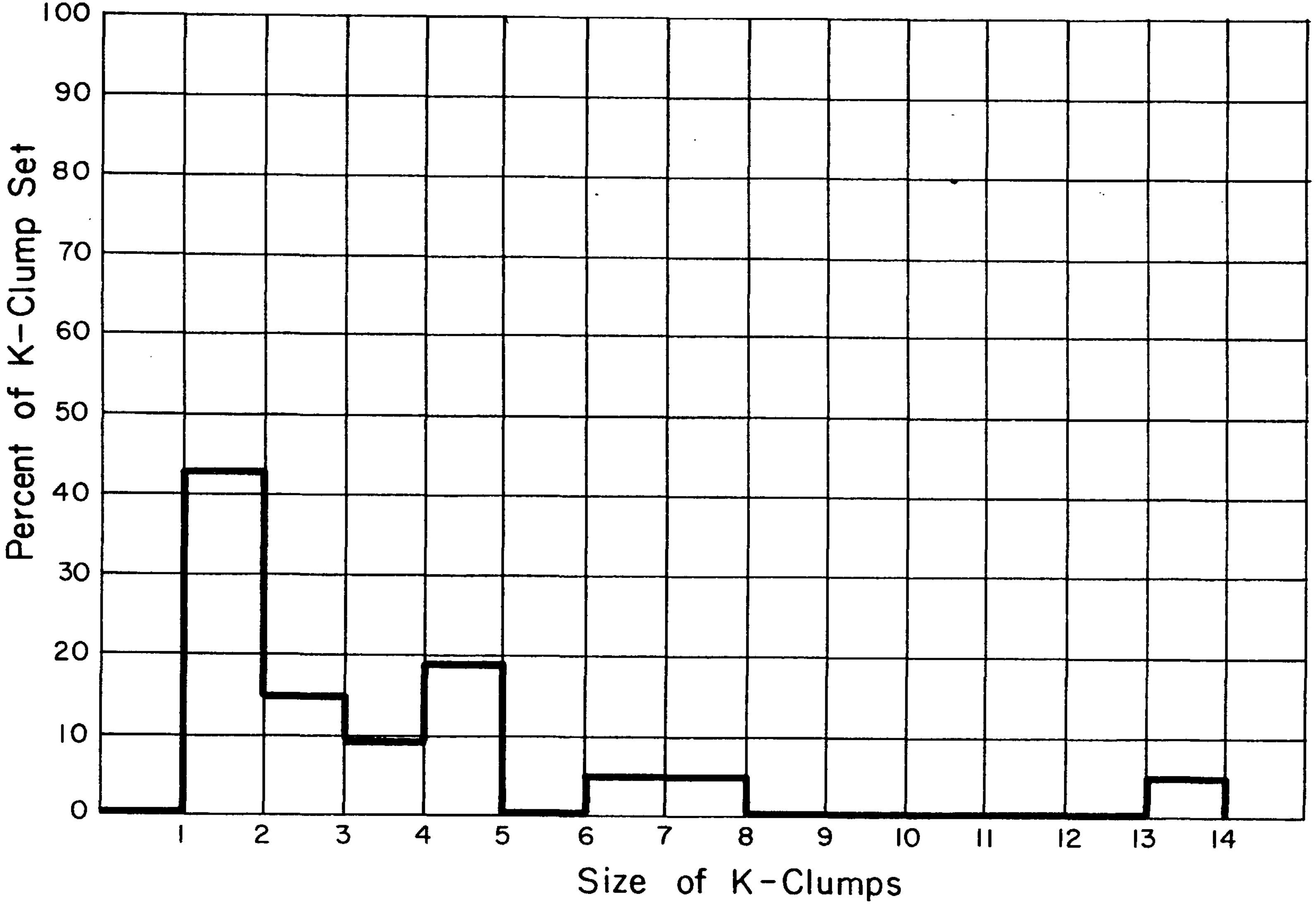
Phase 3:

The connection matrix was computed as described in the experimental design. K-clumps partitioned the set of 84 categories into component sets. The K-clumps ranged in size from 2-14 classes. Graph 3 shows the number of classes by size of the K-clumps.

GRAPH 2: Size of GR-Clumps



GRAPH 3: Size of K-Clumps



REFERENCES

1. S. M. Lamb, "On the Mechanization of Linguistic Learning," University of California, Berkeley, California, 1961.

2. R. J. Solomonoff, "The Mechanization of Linguistic Learning," Zator Company, Cambridge, Massachusetts, 1959.

3. K. C. Knowlton, "Sentence Parsing With A Self-Organizing Heuristic Program," M.I.T., Cambridge, Massachusetts, 1962.

4. K. Sparck Jones, "Synonymy and Semantic Classification," Cambridge Language Research Unit, Cambridge, England, 1964.

5. E. D. Pendergraft, "Basic Methodology," *Symposium on the Current Status of Research*, University of Texas, Austin, Texas, 1963

6. N. Dale, "Automatic Classification System Users' Manual," University of Texas, Austin, Texas, 1964.

7. A. F. Parker-Rhodes and R. M. Needham, "The Theory of Clumps," Cambridge Language Research Unit, Cambridge, England, 1960.

8. C. F. Hockett, *A Course in Modern Linguistics*, New York City, New York, 1958.

9. Z. S. Harris, *Methods in Structural Linguistics*, Chicago, Illinois, 1951.

10. A. G. Dale, N. Dale and E. D. Pendergraft, "A Programming System for Automatic Classification with Applications in Linguistic and Information Retrieval Research," University of Texas, Austin, Texas, 1964.

11. A. I. Khinchin, *Mathematical Foundations of Information Theory*, New York City, New York, 1957.

12. C. F. Hockett, "Linguistic Elements and Their Relations," *Language*, January 1961.

13. R. M. Martin, *Truth and Denotation*, Chicago, Illinois, 1958.

14. W. B. Estes, W. A. Holley and E. D. Pendergraft, "Formation and Transformation Structures," University of Texas, Austin, Texas, 1963.

15. Linguistics Research Center, Report No. 23, University of Texas, Austin, Texas, 1965.

16. M. Joos, "Semology," Summer Institute of Linguistics, University of Texas, Austin, Texas, 1961.

17. S. M. Lamb, "On the Nature of the Sememe," University of California, Berkeley, California, 1961.

18. L. Bloomfield, Language, Chicago, Illinois, 1933.

19. D. A. Senechalle, "Q-Collections and Concatenation," University of Texas, Austin, Texas, 1963.

20. W. R. Ashby, Design for a Brain, New York City, New York, 1960.

21. B. Foster, "Information Maintenance System Manual," University of Texas, Austin, Texas, 1964.

22. A. G. Dale and N. Dale, "Some Clumping Experiments for Information Retrieval," University of Texas, Austin, Texas, 1964.