

AD 609-180

A PROGRAMMING SYSTEM FOR
AUTOMATIC CLASSIFICATION
WITH APPLICATIONS IN LINGUISTIC
AND INFORMATION RETRIEVAL RESEARCH

A. G. Dale, N. Dale
E. D. Pendergraft

prepared for
National Science Foundation
Grant NSF GN-208

— LINGUISTICS RESEARCH CENTER
The University of Texas
Box 7247, University Station
Austin, Texas 78712

LRC 64 WTM-4

October 1964

CONTENTS

Foreword	v
Abstract	vii
1 Introduction	1
2 System-Description	7
2.1 General features.....	7
2.2 Processing details	8
3 System-Summary	13
4 Research Use of the System	15
4.1 Document Retrieval Research	15
4.2 Linguistics Research	15

FOREWORD

This paper was presented originally at the International Study Conference on Classification Research held at Elsinore, Denmark, in September 1964. Research on automatic classification at the Linguistics Research Center is supported by the National Science Foundation and the United States Army Electronics Laboratories.

ABSTRACT

Two key problems, both of which must be solved as a part of continuing progress toward full use of the computer concerned the authors: automatic classification techniques, and how to use them as a basis for self-organizing descriptive systems in syntactic and semantic research.

Under the first, we had three objectives:

1. An appropriate basis for retrieval operations in the machine environment.
2. Techniques that are computationally feasible on a significantly large scale.
3. Immediate use in order to facilitate current research and experimentation with man-machine systems.

Algorithms developed largely by R.M. Needham and based on the theory of clumps provided the techniques for computing the necessary classifications and associations. Moreover, experiments by Needham and others have demonstrated the usefulness of the classification technique as a basis for associative document retrieval.

These considerations have prompted the development of a programming system that provides a flexible computational facility for research and experimentation with classification techniques.

Two applications involving the projected use of the programming system are described.

The first discusses a feedback retrieval model in which clumped associations among index words are used for sample output from a document file and user feedback supplies additional data for a reordering algorithm. The second outlines a proposed procedure for automatic generation of syntactic and semantic descriptions, given primitive symbols of an input text and certain binary relations between symbols as initial data.

The development of the programming system described in this paper has been prompted by several considerations.

- (1) The lack of adequately documented computing algorithms and the difficulty of transferring existing machine-dependent computer programs are obstacles to productive experimentation with automatic classification techniques.
- (2) The need for more extensive experimentation has been recently noted by other investigators [1,2].
- (3) The techniques embodied in the programming system are of interest, since experimentation has indicated that they provide an appropriate basis for linguistic classification and information retrieval operations of a special type [3,4].
- (4) Feasibility studies show that they can be applied on a significantly large scale on computing equipment now becoming available, within economically acceptable processing times. Some of the alternative computer-oriented classification techniques that have been proposed will present severe, and perhaps intractable, problems if attempts are made to extrapolate use to an interesting scale.

The classification procedures in the system are based on algorithms developed largely by R. M. Needham and A. F. Parker-Rhodes [5,6], and provide for the clumping of a universe set of objects into subsets that are homogeneous

in their properties by some criterion. Assuming that a symmetrical relation is defined between pairs of objects so that for each pair x, s we can attach a real positive number $c(x, s)$, called the connection of the pair, the symmetrical matrix of these connections forms the data array for the classification procedures. The following notation will be used in briefly reviewing the algorithms:

U: the universe set

A: a subset of U

B: the complement of A ($U - A = B$)

x: an element of U

a_i : an element of A ($a_1 \dots a_t$)

b_i : an element of B ($b_1 \dots b_r$)

$C(x, A): \sum_{i=1}^t c(x, a_i)$ (i.e. the total connections of an element x to all elements in A)

$C(x, B): \sum_{i=1}^r c(x, b_i)$

$b(x, A): C(x, A) - C(x, B)$. The bias of an element to a set A is the algebraic sum of its connections to A and B (note that bias may be positive or negative).

$AXA: \sum_{i=1}^t \sum_{j=1}^t c(a_i, a_j)$ (i.e. AXA is the summed connections within A, given the convention that $c(a_i, a_i) = 0$)

$BXB: \sum_{i=1}^r \sum_{j=1}^r c(b_i, b_j)$ (i.e. the summed connections within B)

$AXB: \sum_{i=1}^t \sum_{j=1}^r c(a_i, b_j)$ (i.e. the summed connections between elements of A and B)

The first algorithm available within the programming system is a procedure for finding what Needham has termed GR-clumps. A subset, A, of the universe is a GR-clump if all the members of A have positive or zero bias to A, and all of the members of B have negative bias to A. The computing procedure is essentially as described in an earlier paper [7], except that a wider range of options is available for defining initial partitions for the clumping algorithm.

The second algorithm locates subsets that are homogeneous by a different criterion than that used to define GR-clumps, and is based on a procedure suggested to the authors in discussions with Needham. A GR-clump is, by definition, a local minimum for the function:

$$F(A) = \frac{AXB}{AXA + BXB} \quad (1)$$

Accordingly, the process of locating GR-clumps may be considered as a search for local minima of this function. Unfortunately, use of the function of equation (1) is an unsatisfactory search heuristic, since it is clear that its absolute minimum is $F(A) = 0$ for the trivial cases where A is null or is the universe set, and searches for local minima will tend to collapse to this value. Needham therefore suggested investigation of the function:

$$F(A) = \frac{AXB^2}{AXA + BXB} \quad (2)$$

As A approaches the null set or the universe set the value of the function tends to increase rapidly¹, so that searches for local minima will not tend to the trivial cases.

¹ Except in special cases where small disjoint sets exist in U. In current experimentation such sets have been identified by the procedure.

We may employ a process description to define the clumps found by the procedure. Given any partition of the universe into A_i and its complement B_i , let A_{i+1} denote the addition of an element p_i to A_i ($A_{i+1} = A_i + p_i$).

A_k is a D-clump² if:

- a. From some initial partition of the universe into A_1 and B_1 , there exists a sequence of sets, $A_1 \dots A_k$ such that $F(A_i)$ exists for all i ($i = 1 \dots k$).
- b. $F(A_{i+1}) < F(A_i)$, ($i = 1 \dots k-1$).
- c. There is no element p_k such that $F(A_{k+1}) < F(A_k)$.
- d. $A_k X A_k > A_k X B_k$, where B_k is the complement of A_k .

The first three conditions relate to the search procedure and indicate that the terminal sets, A_k , correspond to special minima of the function. Since the procedure does not remove elements from A_i , it cannot be asserted that $F(A_k)$ is a local minimum. The last condition requires the interconnections within A_k to be greater than the cross connections between A_k and B_k , since it is possible to find terminal sets that do not satisfy this intuitively desirable condition.

D-clumps found by the procedure frequently approximate GR-clumps, although their properties are formally weaker than clumps of the latter type. A formal discussion of D-clumps will be the subject of a later paper. Our interest in the procedure has been prompted partly by the possibilities of rapid computation, and partly by the prospects for locating clumps that may be as equally useful for purposes of automatic classification as GR-clumps.

²"D-clump" is an unofficial christening for identification purposes in our own research group.

The connections presupposed in clumping of any type may be defined in terms of a matrix giving the incidence of properties among objects of the universe set U . Since the connections formalize the similarity between two objects through reference to common properties, clumping will tend to associate sets of objects that share or lack certain properties. Thus an alternate technique is suggested in which the properties rather than the objects of the incidence matrix are classified. The properties are regarded as a second universe set U' , and the connections formalize similarity between properties through reference to their coincidence among objects. Clumping in U' will accordingly associate properties that are similar in their patterns of incidence among the objects of U .

Cross-clumping is a technique based on the observation that clumping in the pair of universes U, U' should yield pairs of clumps A, A' which satisfy both of these conditions; viz. the properties of A' are similar in their patterns of incidence among the objects of A , while the objects of A tend to share or lack the properties in A' . Incidence patterns of A' among A will therefore be either dense or sparse by comparison to those of U' among U .

The programming system for automatic classification is written in FORTRAN IV, a machine-independent programming language. It is currently being implemented on an IBM 7040 computing system with approximately 32K words of central memory. The system requires access to three magnetic tape units for temporary input-output storage.

2.1 General features

The main program is an executive routine that reads in parameters specifying the processing options to be executed, and initiates calls to four major program segments. The major segments are as follows:

- MATRIX: A set of programs to read in initial incidence data and form compact connection matrices.
- CLUMP: Separate clumping algorithms of the types described above.
- SORT: A sorting program that scans the output of the clumping algorithms, removes duplicate clumps, and computes the amount of overlap between pairs of clumps in the clump set.
- CROSS: A set of programs that, given clumps of objects and clumps of properties, identifies submatrices of the initial incidence matrix that are significantly sparse or dense.

Each major segment communicates with other segments through standard input-output functions, so that the programming system is entirely self-contained and requires no intervening manual operations after the input of initial data.

2.2 Processing details

Each major segment consists of a segment executive routine that generates a required calling sequence for its subprograms.

MATRIX

The MATRIX package includes seven major sub-routines:

Input routines. These read initial numerical incidence data in various specified formats of the type that will normally be encountered. It is possible to add special-purpose input routines to accept data in formats not compatible with the existing set of routines. Output from this subpackage is a standard binary tape for use in the routines following.

Transposition routine. The subroutines that form the connection matrices assume that initial incidence data has been supplied as rows of an incidence matrix, and that properties will have row designations and objects to be clumped will have column designations. Since this will not always be the way in which initial incidence data will be supplied to the program (i.e. objects to be clumped might have row designations), a transposition routine is included. This will transpose the incidence matrix to provide a correct data arrangement for the subroutines that form the connection matrices.

Connection matrix formation. Four measures of connection (similarity) between pairs of objects may be computed. Each uses a different subroutine. The first three assume a binary incidence matrix (i.e. objects are characterized as possessing a given property, or not possessing it). Given

this type of characterization in an $m \times n$ property-object matrix, let $l(i,j)$ be the number of 1's in the intersection of columns i and j (i.e. the number of properties objects i and j have in common, and $l(i)$ be the number of 1's in the i th column (i.e. the total number of properties defining the i th object):

Connection Def. 1: $l(i,j)$

Connection Def. 2: $l(i,j)$

$$l(i) + l(j) - l(i,j)$$

Connection Def. 3: $l(i,j)$

$$\sqrt{l(i) \cdot l(j)}$$

Each of these measures of similarity between pairs of objects has been suggested in the literature, and can be computed from a given set of appropriate incidence data.

The fourth connection definition is used in cases where the initial incidence data consists of nonbinary attribute values. Given vectors of integer attribute values in the incidence matrix the subroutine (a) normalizes column vectors by column totals, and (b) computes associations between pairs of objects as follows:

$$\text{Connection Def. 4: } \frac{1}{\sum_{k=1}^m (a_{ik} - a_{jk})^2}$$

where a_{ik} and a_{jk} are the normalized values on the k th property for the i th and j th objects.

Each connection matrix is compacted, in the sense that only nonzero elements of the full $n \times n$ matrix are recorded. The clumping algorithms are programmed to work

with connection matrices that are stored within the central high-speed memory; compaction is desirable to permit work with as large a universe set as possible. On the IBM 7040, approximately 20,000 central memory locations are available to hold connection values during the clumping process, so that there is a physical restriction to connection matrices with this number of nonzero elements. In an information retrieval context where index words are clumped to produce word associations, this permits work with vocabularies of the order of 750-1000 words, since the connection matrices will typically have densities of the order of 2-3 per cent.

Output routines. These are control sequences that write the connection matrix on magnetic tape if the user wishes to save it for future processing.

CLUMP

The CLUMP package consists of four major sub-routines:

Input routine. This reads in the connection matrices if they have not been left in core by the previous segment.

GR-clump algorithm.

In using this algorithm, options are available in the choice of initial partitioning procedures. The user can select as initial partitions: (1) sets defined by the pivot variable method (described in [7]); (2) random initial partitions of $n/2$ elements; (3) K-clumps³; or (4) terminal sets found by the D-clump algorithm.

D-clump algorithm (described above). In its present form this algorithm takes K-clumps as initial sets.

³ Given a connection threshold, T , a K-clump is a set of elements such that each pair in the set has connection greater than T , and no element not in the set has a connection greater than T to each member of the set. See [6].

K-clump algorithm. This routine is used to produce starting sets for the two major clumping procedures.

SORT

The control program for post-clumping sorting and checking has routines that sort clumps found by the algorithms and eliminate duplicate clumps. The sorted clumps are stored on magnetic tape. An analysis of the clump set is made, producing as output a table showing the amount of overlap in the set of clumps.

CROSS

Assume that both objects and properties defined in the original incidence matrix have been clumped (i.e. that both row elements and column elements have been considered in turn as objects for clumping). Then the i th clump of row elements defines a subset of rows in the original incidence matrix, and the j th clump of column elements defines a subset of columns in the incidence matrix. Together the two clumps define a submatrix of the incidence matrix. It is desired to characterize such submatrices as either sparse or dense, depending on whether or not the ratio of nonzero elements in a submatrix is significantly less than or greater than the average density of the entire incidence matrix.⁴ This can be treated as a classical sampling problem involving testing the null hypothesis at desired confidence levels. Given m clumps of row elements and n clumps of column elements, it is possible to construct an $m \times n$ table characterizing the submatrices defined by all pairs of the clumps.

⁴It is assumed that the incidence matrix is composed of binary values. Non-binary incidence matrices may be converted to binary form by defining a suitable threshold value, T , and setting an element, a_{ij} , = 1 for $a_{ij} \geq T$, and $a_{ij} = 0$ for $a_{ij} < T$.

The CROSS subprogram contains subroutines that:

- (1) form and examine the appropriate submatrices, characterizing them as sparse or dense at 99 per cent or 95 per cent levels of confidence (at the option of the user); and
- (2) forms the complete $m \times n$ classification array based on this information.

SYSTEM-SUMMARY

The main features of the programming system are:

- (1) It is written in a machine-independent programming language.
- (2) It permits great flexibility in the use of processing options - a desirable characteristic in a research environment.
- (3) Its use involves simple operating procedures, since processing options are initiated through a sequence of simple control cards. Accordingly, investigators unfamiliar with computing procedures should be able to use the system with a minimum of difficulty.
- (4) The system has been tested with an incidence matrix of approximately 1000 X 350, clumping the 350 column elements. Approximate execution times for the major segments on an IBM 7040 (8 μ s core cycle time) are as follows:
 - a. Formation of initial connection matrix (Def.1): 70 minutes (incidence data input from magnetic tape). The connection matrix has 6.7 per cent nonzero elements. Connection matrix formation time is a nonlinear function of matrix density and dimension.
 - b. Formation of Def. 2 and 3 connection matrices: \sim 10 seconds each, from the Def. 1 matrix.
 - c. D-clumping: \sim 7 seconds per trial (including time required to generate K-clumps as initial partitions).
 - d. GR-clumping: \sim 6 seconds per trial.

4 RESEARCH USE OF THE SYSTEM

Two uses proposed for the system in connection with research at the Linguistics Research Center during the coming year are briefly discussed below.

4.1 Document retrieval research

Use of remote input-output terminals connected with a central computing system is planned, to permit experimentation with rapid document retrieval from files maintained in the computing system. Terminal use will be time-shared on the central computer, permitting simultaneous operating of terminals, and, possibly, simultaneous searching of several document collections. Apart from a capability for rapid ("real-time") retrieval, the system will permit experimentation with the use of user feedback information to direct the retrieval process. The basic retrieval algorithm has been described in an earlier paper [4]. It uses index word clumps as a basis for document classification and for associative retrieval, and an experimental feedback algorithm has been specified. The programming system described will be used to permit experimental retrieval with different clump and connection definition with a document collection of approximately 1500 items, indexed from a vocabulary of 700-1000 words.

4.2 Linguistic research

Experimentation with self-organizing linguistic systems will make use of the classification algorithms in combination with the automatic syntactic analysis and synthesis algorithms of the Linguistics Research System (LRS) [8,9]. As an initial research objective, an attempt will be made to improve a description of English syntax through

automatic syntactic analysis and classification of an extensive English corpus. The status of the syntactic description at various phases of classification will be sampled by means of randomly synthesized English expressions.

Cross-clumping was developed primarily for these linguistic applications. Here the universe sets U and U' may be regarded as the domain and counterdomain of some binary relation R . The membership in R of pairs x, y is given by the incidence matrix.

Linguistic segmentation can be described by means of the binary relation of concatenation so that, given incidence data of the form $x \hat{y}$, cross-clumping will find pairs of clumps $A \hat{A}'$ whose members tend to concatenate. Concatenation incidence data will be compiled mechanically from the results of automatic syntactic analysis. The clumps A and A' will be represented in U and U' by the addition of elements a and a' (signifying an instance in the corpus of any member of A or A') respectively. Hence iterative analysis and classification will discover structural relations of the types $x \in A$, $y \in A'$ or $A'' \supseteq A \hat{A}'$ describable by syntactic rules (e.g. the phrase structure rules $a \rightarrow x$, $a' \rightarrow y$ and $a'' \rightarrow a \hat{a}'$).

These inductive operations will usually produce syntactic categories that are too general. Syntactic subclassification can be mechanized, however, by interpreting the relation R as that of application [10] between the syntactic rules. Application of the rule $a \rightarrow x$ to the rule $a'' \rightarrow a \hat{a}'$, for example, will result in the derived rule $a'' \rightarrow x \hat{a}'$.

Potential applications involving the category a are therefore given by the domain U of the instances of

the symbol "a" as the left member of a syntactic rule, and by the counterdomain U' of instances of the symbol "a" in some position of the right member of a syntactic rule. Non-binary incidence data, recording the frequency of application for the various pairs of instances, will be compiled mechanically from results of automatic syntactic analysis. Each pair A,A' of clumps discovered by cross-clumping will define a subcategory b of category a. The subcategory will be incorporated in the syntactic description by a process which duplicates the rules involving the instances of the symbol "a" in A and A' and replaces these instances by another symbol "b".

The problem of overclassification within the self-organizing system will be approached as follows:

- a. Subclassification with a particular category will be inhibited when (a) inadequate incidence data have been collected, or (b) the entropy of the normalized connections among the elements has exceeded a given parameter. A category whose entropy has reached status (b) will be regarded as stable or residual according as the average value of connections among the elements is less or greater than the value of a parameter.
- b. Stable categories that have essentially the same membership will be identified. This will be attempted by taking the set of all stable categories as the universe U, and collecting data on the coincidence of such category representations in the syntactic ambiguities resulting from automatic analysis. Every clump A in U will be

recorded as a distributional attribute of each category in A. The distinction between categories with identical (or very similar) attributes will be eliminated.

- c. Residual categories will be reduced by eliminating some of the syntactic rules which describe them. If b is a residual category, for example, then any rule (such as $b \rightarrow x$ or $b \rightarrow a \wedge a'$) having the symbol "b" as the left member will be eliminated when empirical probabilities reflecting use of that rule in analysis fall below specified values.

Analogous techniques will be applied to the self-organization of semantic descriptions within LRS. Here the syntactic rules generated by the system will be taken as the objects to be classified. Furthermore, since all of the procedures are generalized as to language, they will be used on languages other than English whose descriptions are currently maintained in LRS, especially German, Russian and Chinese.

REFERENCES

- 1 L. B. Doyle, Sixty Ideas in Sixty Months, Systems Development Corporation, Santa Monica, June, 1964.
- 2 M. Kochen, Some Problems in Information Science with Emphasis on Adaptation to Use Through Man-Machine Interaction, International Business Machines Corporation, Yorktown Heights, April, 1964.
- 3 R. M. Needham and K. Sparck Jones, "Keywords and Clumps," Journal of Documentation, vol. 20 No. 1 March, 1964
- 4 A. G. Dale and N. Dale, Clumping Techniques and Associative Retrieval, NBS/ADI Symposium on Statistical Association Methods for Mechanized Documentation, Washington, March, 1964.
- 5 A. F. Parker-Rhodes and R. M. Needham, The Theory of Clumps, Cambridge, 1960.
- 6 R. M. Needham, The Theory of Clumps II, Cambridge, 1961.
- 7 A. G. Dale and N. Dale, Some Clumping Experiments for Information Retrieval, Linguistics Research Center, Austin, February, 1964.
- 8 Linguistics Research Center, Machine Language Translation Study (contracts DA 36-039 SC 78911 and AMC-02162(E) and Development of a Linguistic Computer System (grants NSF G-19277, GN-54 and GN-208) Progress Reports 1-20, 1959-64.
- 9 E. D. Pendergraft, The Linguistics Research System, Linguistics Research Center, Austin, September 1964.
- 10 W. B. Estes, W. A. Holley and E. D. Pendergraft, Formation and Transformation Structures, Part I, Linguistics Research Center, Austin, May 1963.