A COMPUTER MODEL FOR CERTAIN CLASSES OF VERBAL BEHAVIOUR

G. R. Kiss

Medical Research Council
Speech & Communication Research Unit
Edinburgh

Abstract

A computer model of some classes of verbal behaviour is described. The general characteristic of the classes is that the subjects emit discrete verbal units but not continuous discourse. The concept of a word store is defined as a stochastic information retrieval system, implemented in the form of a network of stochastically interacting nodes. The word store is assumed to be a parallel processor which operates in conjunction with a serial processor in the human organism. The application of the model to word association, response sequences, learning processes, and satiation effects is discussed. The main data structure and functions of the computer program are described with some examples given in the POP-2 programming language. It is shown that the word store model fits qualitatively many of the phenomena in verbal behaviour, and an example of a quantitative comparison is shown.

Descriptors

Words, memory organisation, retrieval, association, verbal behaviour, computer model, stochastic, network.

Introduction

This paper gives a brief description of a set of computer programs which constitute a model of human behaviour in certain experimental situations like word association, free recall of word lists, paired associate learning, and others. The general characteristic of all these experimental settings is that the subject is required to select and emit discrete verbal units, but not continuous discourse.

In spite of the great deal of activity in the field of verbal behaviour, there is a surprising lack of general theories which could account for the phenomena in a systematic manner. The model described in this paper is a tentative step in this direction. Various aspects of this model have been discussed in a number of other papers[12,13,14,15] and some of these will be summarised very briefly here. The main emphasis will be on the behaviour and potentialities of the computer simulation.

Some ideas about the need for parallel processing in cognitive processes have also been discussed by Reitman[20], and his approach is close to that of the present paper. Feigenbaum's EPAM model provides an example for a rather different approach to the computer simulation of verbal behaviour.[6] Finally, Quillian[19] is working on another more elaborate computer model for semantic memory which is again rather relevant to many of the ideas in this paper.

Formulation of the Model

The general assumption is made at the outset that humans possess both serial (central-processor like) and parallel information processing resources, and that these are jointly utilised in high-level psychological processes. In order to deal with verbal behaviour it is postulated that the relevant parallel processor (which will be called the word store) is functionally a stochastic information retrieval system. Structurally it is a network containing representations of words and links between them. Each word (representation) can have a varying level of activity. The links can transmit activity from one word to another.

This transmission is assumed to be stochastic in nature and is described by specifying the probability that a unit activity in word ^ will produce a certain distribution of activities over the words in the system. It is argued in Kiss,[15] however, that in normal behaviour the interest is in the mean values of these transmissions which are, due to the distributed manner in which the links are implemented in the organism, rather stable and can be treated in a deterministic manner.

Each word is also capable of storing the cumulative sum of activities received over an interval of time. This sum will be referred to as the cumulative activity of a word. It is assumed that this activity either persists indefinitely, or decays according to some (possibly exponential) function.

The word store can now be looked at

as a sequential machine whose state is defined by a vector specifying the current cumulative levels of activities of the words. The system can make transitions in the absence of external influences (free transitions) due to the operation of the transmission links. External interference with the activities can also change the state of the system. The fine behaviour of the word store can be described by the machinery of stochastic multidimensional branching processes.15 The average behaviour, on the other hand, can be treated by a graph theoretical approach, in particular by using signal flow graphs.14

The interaction between the word store and the serial processor is defined as follows. At any instant a word can become available to the serial processor ("can enter consciousness") with a certain probability. This probability is the ratio of the cumulative activity in that word to the sum of all such activities in the word store. This is in accord with some well-known theories of choice behaviour (e.g. Luce16).

In summary, the two essential components of the word retrieval process are the evolution of the word store in time through transitions from some starting state, and a "decision stage" in which a random choice is made according to the relative levels of cumulative activity.

Verbal Behaviour and the Word Store Model

Word Association

The word association experiment is perhaps the most uncluttered form of verbal behaviour. In its simplest form the subject is minimally constrained by giving a stimulus word, and any response is acceptable. Nevertheless, when the frequencies of response words are tabulated for samples of subjects, surprising regularities emerge. The frequency distribution has a characteristic shape, shown to be of the Zipf type by Skinner.22 Response latency depends on response probability as a smooth monotonic decreasing function, thought to be logarithmic.24 The distribution of latencies is close to the lognormal.8,12 The absolute probability of a word in the language correlates highly with its average associative probability.8 The rank of a word in the distribution is positively correlated with the number of other words in the response list which also elicit that word as a response.18

This is a very incomplete list. Numerous other findings are summarised ir a recent book by Cramer.4 Those listed above are most closely related to the model and hopefully comprise a fairly representative selection of phenomena to be explained. Let us outline briefly how the word store model accounts for them.

One particularly attractive way of conceptualising the operation of the model is to think of it as a signal flow graph. This is a linear directed graph in which the nodes are words and the arcs are the transmission links. Every node adds the incoming signals and transmits the result along the outgoing arcs. The values of the arcs are the transmittances connecting one node to another. By usin a basic set of elementary transformations which deal with series, parallel, star, and self-loop connections of arcs, one ca calculate the total transmittance between any node pair of a network.14 If the values of the outgoing arcs are normalise so that they sum to one, the representation of a Markov chain is obtained. The n-step transition probabilities, and the sums of n-step transition probabilities u to some limit correspond respectively to the activity transmitted from one node to another at the ntn transition of the system, and to the cumulative activity transmitted during the n transitions. For simplicity, it is assumed here that the word store operates in discrete time: the states of all nodes are updated simultaneously during each transition.

It is possible to assess the values of the arcs in this flow graph by means o the word association experiment, but this is not simple.15 The problem is that th empirical associative probabilities are not the one-step transition probabilities of the Markov chain described above. Bu it is possible to obtain approximations from this data.

The word association process can now be described as follows. Some activity is injected into the node corresponding to the stimulus word in the network. Th word store then goes through several transitions. During each of these the nodes sum the incoming transmissions and transmit the sum to other nodes to which they are connected. At some stage, which depends on the experimental conditions, a choice is made among the words in such way that the probability of choosing any one word is determined by the relative activity of that word.

If this model is used for the simulation of the word association behaviour of a sample of subjects, a number of qualitative and quantitative conclusions can be reached. If the assumption is made that the transition rate is the same for all subjects, then the response latency will depend on the number of transitions made by the word store. If we also assume that the network structure is the same for all subjects (clearly an over-simplification), then associative response probability corresponds to the total transmittance between a node pair, taking into account all possible pathways up to some limiting length. Then the monotonic decreasing function relating latency to probability, described by Woodworth and Schlosberg24 follows from the model, since activity will build up more quickly at the nodes which have a strong total transmittance leading to them from the stimulus node.

The response probability distribution has been obtained by simulation and is compared with some independent empirical data in Figure 1. The agreement is close in spite of the approximations involved in deriving the network data (described in more detail15). The latency distribution cannot be accounted for by the model unless further assumptions are made about the stopping point of the branching process. It seems reasonable to assume that the process stops when some threshold conditions are reached in the activity levels. The various possibilities have not been explored as yet in detail. It is worth pointing out that Williams23 has shown that for a one-dimensional branching process the first-passage time distribution is very close to lognormal.

The result obtained by Pollio18 about the relationship between response rank and the number of other items in the response list eliciting the word is obvious from the model: the more numerous the pathways between stimulus and response, the larger the total transmittance and hence the response probability.

Finally, we reproduce here Table 1 from Kiss15 to show the comparison between simulation and empirical data. The column marked ps shows the simulated probabilities, and the columns marked PM1 and PM2 show the probabilities obtained from 1,000 subjects at Minnesota in 1954 and 1964 respectively. A star in the pm2 column means that the probability is not known, only the fact that the word does occur (this is due to the peculiarities of the tabulation in the Minnesota data, see Palermo and Jenkins17). It can be seen that 80 per cent of the words retrieved by the model are in the Minnesota data. The correlation between the simulation and the Minnesota data is r=0.578. This simulation involved letting the word store run through three transitions. Similar comparisons have been made when the stimulus consists of three words, for example moth bird fly.12 The difference here is that a starting state is set up in which several words have nonzero activities. The correlation between the empirical and simulated probabilities was r=0.73.

Response Sequences

In many experiments the subject is required to emit a sequence of responses. Responses which are acceptable can be specified in a number of ways. Bousfield and Sedgewick1 asked subjects to generate members of some class, like four-legged animals, names of cities, etc. Less restrictively, the subject may be asked to generate a continuous sequence of word associations to a stimulus word. One may also consider the free recall of a word list as a highly restricted case of this process.

How does the word store model cope with this situation? It is assumed that there are two components, carried out by the parallel and serial processors respectively. The parallel processor is generating candidate items which are then tested for acceptability by the serial processor.12,15 The parallel processor generates the candidates by adjusting the probabilities with which words become available to the serial processor, in the manner described in the previous section. Particular questions which arise here are the setting up of a starting state for the system, and the feedback to the system from the testing and emission of candidates.

The characteristic results of the Bousfield and Sedgewick1 experiments and of the continued association experiments are that if the cumulative number of responses given is plotted as a function of time then the average curve is well described by an equation of the form n=c(l-e-mt). When the fine structure of individual curves is examined, however, one finds that the responses occur in bursts, triggered by a member of some sub-class, like domestic animals, etc. A related phenomenon occurs in the free recall of word lists where the items show a positional clustering even when the presentation order of the list is random.

In attempting a computer simulation

of such sequential phenomena the assumption was made that the emission of a response by the subject has the same feedback effect as the presentation of a stimulus, possibly with an attenuation parameter reflecting the "attention" paid to the event. Similarly, the testing of an item by the serial processor is assumed to have a feedback effect, even if the item is rejected as unsuitable. Once these assumptions are made, the same program could be used here as for the generation of single associative responses. It was necessary to add only a few trivial executive routines in order to generate continued associative sequences.

In the simplest case the feedback effect was set equal to the stimulus effect and a feedback injection of activity was made whenever a candidate item was tested, irrespective of the outcome. The only test in this case was for previous occurrence: if the item was new it was emitted, if not it was rejected. The average curve for 10 runs is shown in Figure 2. It is quite clearly of the kind usually observed in these experiments .

More impressive was however the clear-cut occurrence of temporal clustering effects in the individual sequences. An example is shown in Figure 3. Notice that the clusters seem semantically coherent, and that this effect did not require the introduction of 'conceptual organising factors' into the production process advocated by some investigators (see, for example, Cofer3). No detailed exploration of the effects of the parameters has been done at the time of writing this paper.

Learning Processes

No model of verbal behaviour could be complete without the incorporation of learning mechanisms. In this section we shall show qualitatively that the word store model can cover tnem in a natural manner. The detailed exploration of these facilities is in progress.

There is a vast variety of experimental paradigms for verbal learning. Lack of space precludes any extensive discussion of the findings, so that again a set of interesting examples will be taken: paired-associate learning (see, for example, Goss and Nodine7), mediated transfer of learning10 and semantic generalisation (see, for example, Feather5).

Let us start with an exploration of possible loci for learning effects in the word store model. The most obvious possibility is the adjustment of the network structure and of the link values. As far as the adjustment of link values is concerned, the scheme closely resembles the Perceptron mechanisms.21 It should be noted however that the choice of output (monitoring) nodes also contributes to the adaptive capabilities. It seems likely that a hierarchical system of monitoring nodes will be necessary in order to reflect the current state of the system.

These proposals result in a distributed storage of the effects of learning. The scheme which is being explored at present is as follows. In a paired-associate learning situation the stimulus node is activated, then the system is allowed to run for a number of transitions. This results in an activity pattern over a certain area of the network. The nodes with the highest activity levels are selected, down to some threshold level. It is now ascertained whether these nodes are connected directly to the desired response node or not. If a connection exists, it is reinforced; if it does not, then a new connection is built.

Clearly, a number of alternative schemes are possible and the eventual choice will have to depend on detailed comparisons with empirical data. It is felt, however, that the model has a qualitative "fit". For example, a well-known effect in paired-associate learning is that ease of learning increases with increasing m values of the responses. The m value is usually defined as the number of associations a subject can give to a word in a specified time interval. In terms of the word store model m will be a function of the network connectivity in the neighbourhood of the word in question. (To obtain more precise results one should really work with some graph-theoretically defined "sink" or "source" property of a node.) It intuitively follows that, given the learning scheme described above, ease of learning will increase if there are many existing pathways leading to the response term.

Similar remarks can be made about mediated transfer, where the effects of existing associative connections on the transfer of learning manifest themselves. A criticism which can be directed towards most of these experiments is that they fail to recognise the fact that the associative connections are embedded in a network, so that many alternative pathways are operating simultaneously. This may be one reason why the exploration of a number of mediation paradigms10 leads to unclear results.

Semantic generalisation, where a conditioned response is observed to generalise from one word to others which have similar meanings, is a natural property of the word store model if one observes that semantically related items form "strongly" connected clusters in the network.11 According to the learning scheme described, the associations are built in a distributed fashion between some *areas* of the network and the response; hence if semantic clusters exist, semantic generalisation would follow.

The possibility of introducing inhibitory links has not been mentioned so far. Physiologically this is a very plausible mechanism. Although the computer model was originally designed to work with excitatory links only, it was found that both negative links and negative activity values can be dealt with without changing the program. Inhibition will almost certainly be needed to ensure selectivity of response and to prevent undesired generalisation.

Satiation Effects

We shall close this section of the paper with a brief discussion of satiation effects. When a word is repeated or inspected for a prolonged period of time, the subject usually reports that the word loses or changes its meaning. If after such treatment a word association test is made, then the number of uncommon or irrelevant associations obtained increases.

In the word store model repeated presentation of a stimulus has the effect that activity will spread into more distant regions and hence more distant (uncommon and irrelevant) responses become likely. Since the cumulative activity in the network is normalised, the eventual limiting situation is a more or less flat distribution over a large area. The subjective changes in meaning seem to imply that there is an awareness of the overall state of the network in the subject and that the correct cognition of meaning depends on the presence of a characteristic pattern of activities. This is the way in which connotative meaning can be represented in the word store model.

The Computer Programs

The POP-2 Programming Language

The programs simulating the word store model are all written in POP-2. This is an on-line conversational programming language in use at the Department of Machine Intelligence and Perception at Edinburgh University.2 It is a powerful and sophisticated language for non-numerical computation with a variety of data structures, including words, lists, records, strips, and arrays; and with powerful facilities for manipulating functions. The author was previously programming in IPL-V and found POP-2 a considerable advance, even apart from its conversational availability.

The features which this author found most useful are the combination of LISP-type list processing functions and the ability to make assignments; the stack on which the arguments and results of functions are placed; the ability to manipulate functions as data; records; good readability of programs; and automatic store management (garbage collection).

Data Structure of the Model

The main data structure is held in a variable called *memory*. It is a list of records. Each record represents a word (a node in the network) and consists of six components. The components can be accessed by means of the standard updater-selector doublet functions of POP-2. The names of these are: *name*, which handles a character string (the word which is represented by the node); *alist*, which handles the list of links emerging from the node, and their values (strengths); *activity*, which handles the numerical value of the current activity of the node; sjumact, which handles the cumulative activity; and *newact*, which handles the activity received by the node during the current transition of the system. One further component of the record is used in addressing the disc store with large networks.

The component *alist* is a list of pairs. The first element of the pair is a pointer to a node, and the second is the numerical value. If the network is large, only the records are kept in core store and the *alist*1s are stored on disc. Before each updating cycle (see Figure 4) the *alist*'s of the nodes which have a nonzero activity are brought into core store, and all other *alist*'s, if any, are erased. In this way only the currently active nodes have association lists attached to them and core store demands are kept low.

Main Functions of the Model

The central function is the main updating cycle of the network, called *cycle*. This function works through the memory list and for each node it multiplies the activity of the node with the strengths of the links on the *alist*. and adds the result to the contents of the newact's of the nodes to which the links go-

To illustrate the flavour of the language, we give in Figure 4 the function itself, exactly as typed on a Teletype, apart from the underlining which is included here for readibility. The underlined words are syntax words and cannot be used as identifiers. The standard functions lid and tl select the head (LISP car) and tail (TTSP cdr) of a list. The horizontal arrow is the assignment sign. Apart from the usual bracketed functional notation, a.limited form of inverse Polish notation is available by means of the dot operator. For example, m.tlstl(m) takes the tail of the list m. Most of the other features should be self-evident.

When the end of the *memory* list is reached, the function *transfer* is called. It adds the contents of *newact* to *sumact* (updating the cumulative activity), transfers the *newact* into the *activity*, and clears the *newact*, for each node.

The two functions, *cycle* and *transfer*, together define a transition of the system. Most of the other functions are executive routines, generating one or another kind of behaviour, or utility routines for building, modifying, reading and printing network structures.

As an example, the routine *asequence*, used for the generation of continued associations, operates as follows.

```
   Request a stimulus from the console.
   Clear the activities and cumulative
     activities.
   Set the activity of the stimulus to
     1.
   Set the time counter to 0, and
     initialise a response list.
1. Drive the memory through 3 transi-
     tions.
   Normalise the cumulative activities
     to 1 over the memory.
   Make a random choice weighted by the
     cumulative activities.
   Look up the selected response on the
     response list:
     -if it is on the list, go to 2;
     -if it is not, then put it on the
       list, print the name of the
       response, followed by the time
       count;
2. Clear the activities.
   Set the activity of the response to
     1.
   Increment the time count, print a
     newline, goto 1.
```

The main function for modifying the network is *associate*, which takes the names of two nodes as arguments and build a link with a specified value between them. If the link already exists, the old value is replaced by the new one. If one or both of the nodes are new, then fresh nodes are created and the new names are assigned to them. The link is then inserted normally.

A useful function in the learning processes is *alink*, which is a doublet for accessing or updating the value of a link between two specified nodes. In POP-2 a doublet is a function which can occur either as the source or the destination of an assignment. Thus, alink(x,y)+z selects the value of the link between the nodes x and y and places it into z, while zalinE(x,y) updates the value of the link between x and y to be *z*. The updater part of the function *alink* will also create a new link if it does not yet exist.

References

(1) Bousfield, W.A. & Sedgewick, C.H. (1944) An analysis of sequences of restricted associative responses, *J. gen. Psychol*., 30, 149-165.

(2) Burstall, R.M. & Popplestone, R.J. (1968) *POP-2 Reference Manual*, London, Oliver and Boyd.

(3) Cofer, C.N. (1967) Does conceptual organization influence the amount retained in immediate free recall? In: Kleinmuntz, B. (ed) *Concepts and the Structure of Memory*, New York, Wiley.

(4) Cramer, Phebe (1968) *Word Association*, New York, Academic Press.

(5) Feather, B.W. (1965) Semantic generalization of classically conditioned responses: A review, *Psychol. Bull.*, 63, 425-441.

(6) Feigenbaum, E.A. (1961) The simulation of verbal learning behavior, *Proceedings of the Western Joint Computer Conference*.

(7) Goss, A.E. & Nodine, C.F. (1965) *Paired-Associates Learning*, New York, Academic Press.

(8) Howes, D. (1955) Statistical properties of the word-association experiment (abstract), *Amer. Psychologist*, 10, 437.

(9) Howes, D. (1957) On the relation between the probability of a word as an association and in general usage, J. abnorm. soc. Psychol., 54, *75-85.*

(10) Jenkins, J.J. (1963) Mediated associations: paradigms and situations, In: Cofer, C.N. and Musgrave, B.S. (eds), Verbal Behavior and Learning, New York, McGraw Hill.

(11) Kiss, G.R. (1965) Clustering of words in association networks and in free recall, (abstract). Bull. Brit. Psychol. Soc. 20, No. 66 p. 6A.

(12) Kiss, G.R. (1967a) A test of the word selection model using multiple-word stimuli in word association, MRC Speech & Communication Research Unit, Report SCU/7/67.

(13) Kiss, G.R. (1967b) Networks as Models of word storage, In: Collins, N.L. and Michie, D. (eds), Machine Intelligence 1, London, Oliver and Boyd.

(14) Kiss, G.R. (1968) Words, associations and networks, J. verb. Learn. verb. Behav., 7, 707-713.

(15) Kiss, G.R. (1969) Steps towards a model of word selection (in press), Michie, D. and Meltzer, B. (eds), Machine Intelligence 4, Edinburgh University Press.

(16) Luce, R.D. (1959) Individual Choice Behavior, New York, Wiley.

(17) Palermo, D.S. and Jenkins, J.J. (1964) Word Association Norms, Minneapolis, University of Minnesota Press.

(18) Pollio, H.R. (1963) A simple matrix analysis of associative structure, J. verb. Learn, verb. Behav., 2, 166-165.

(19) Quillian, M.R. (1966) Semantic Memory, Report No. 2, Contract No. AF19(628)-5065, Bolt Beranek and Newman Inc.

(20) Reitman, W.R. (1965) Cognition and Thought. New York, Wiley.

(21) Rosenblatt, F. (1962) Principles of Neurodynamics. Washington D.C., Spartan Books.

(22) Skinner, B.F. (1937), The distribution of associated words, Psychol. Rec, 1, 71-76.

(23) Williams, T. (1965), The basic birth-death model for microbial infections, J. Roy. Stat. Soc. Ser. B, 27, 358-366.

(24) Woodworth, R.S. and Schlosberg, H. (1955) Experimental Psychology, New York, Holt."'^

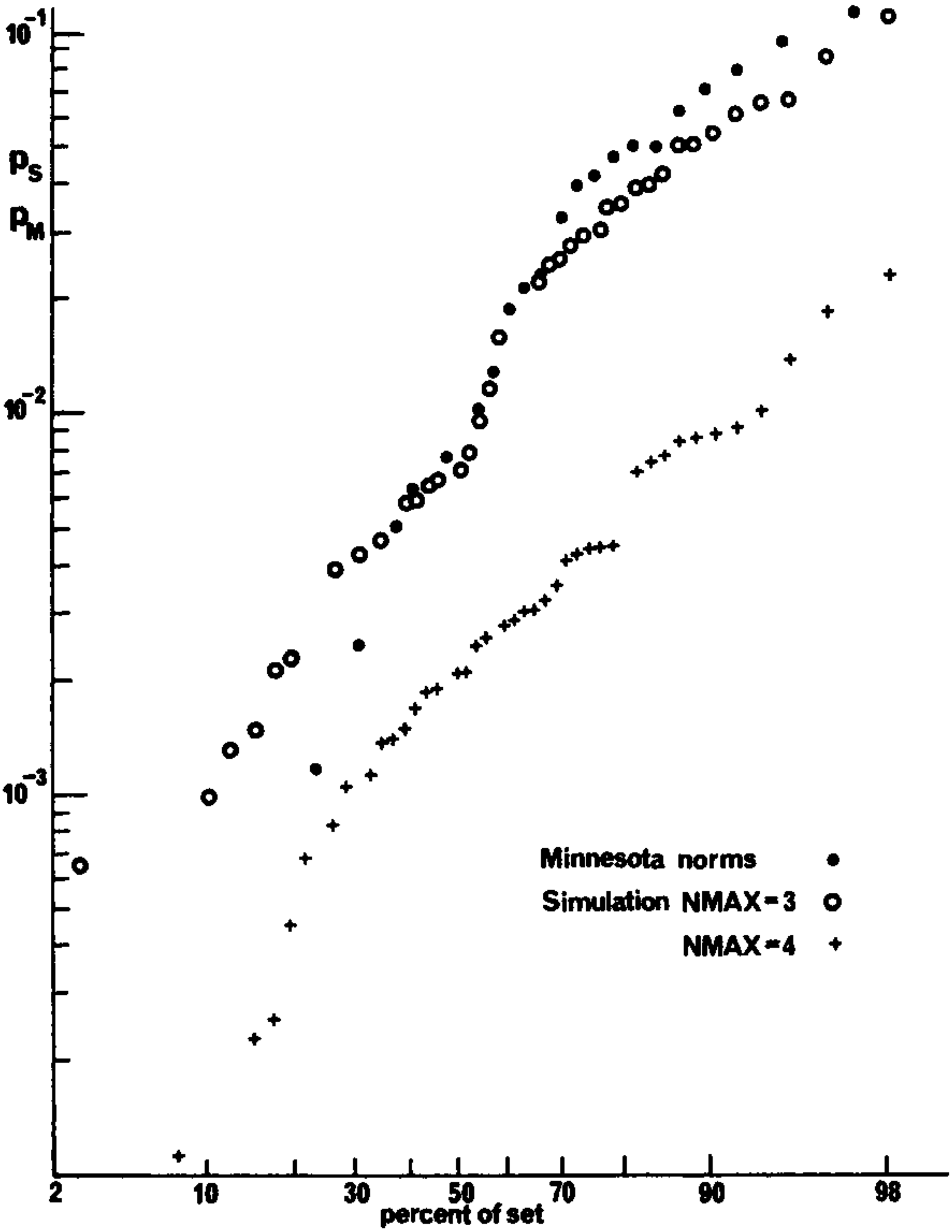

Figure 1. Cumulative response probability distribution on lognormal coordinates. The empirical data is from Palermo and Jenkins (1964). The simulation data is given for 3 (NMAX=3) and 4 transitions of the system. The NMAX=4 curve has been shifted to avoid overlap; its mean is about the same as that of the other curve.

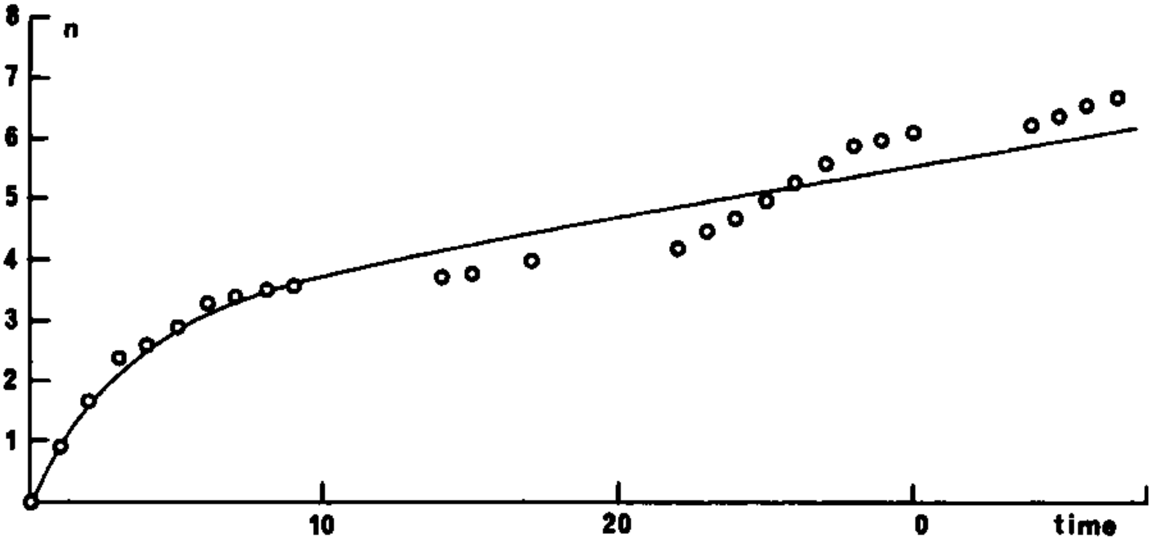

Figure 2. Cumulative number of words emitted as a function of time in a continued association sequence. The curve is the mean of 10 sequences.

```
.ASEQUENCE;
STIMULUS: BUTTERFL
MOTH  0
BUTTERFL  1
NET  2
BEES  3

INSECT  7

FLY  9

LIGHT  11
BIRD  12

WING  16

FLOWER  33

SUNSHINE  35
SKY  36

BLUE  53

COLOUR  65

YELLOW  88
```

```
.ASEQUENCE;
STIMULUS: BUTTERFL
BIRD  0
FLUTTER  1
MOTH  2
WING  3
FLY  4

INSECT  8

BEES  10
FLOWER  11

GARDEN  16

SUMMER  18
CATERPIL  19

YELLOW  37

BLUE  44

SKY  46

COLOUR  53

COCOON  66

BUTTERFL  70
```

```
.ASEQUENCE;
STIMULUS: BUTTERFL
COCOON  0
NET  1
MOTH  2
BUTTERFL  3
FLY  4
WING  5
BIRD  6
BEES  7
FLOWER  8
INSECT  9
GARDEN  10

YELLOW  13
BEAUTIFU  14

PRETTY  18
SOFT  19
```

Figure 3. Examples of association sequences. The numbers after the words are the time counts when the words are retrieved. Note the temporal clustering of semantically related items. Normally the program outputs a "newline" at each time count. The scale has been compressed in the first two sequences to fit the page.

```
function cycle;
vars m x y;
         memory→m;
label1:  if m.null then .transfer exit;
         if activity(m.hd)=0 then m.tl→m;  goto label1 close;
         activity(m.hd)→x;
         alist(m.hd)→y;
label3:  if y.null then m.tl→m;  goto label1 close;
         x*y.hd.tl+newact(y.hd.hd)→newact(y.hd.hd);
         y.tl→y;  goto label3
end
```

FIGURE 4.

TABLE 1. RESPONSES TO THE STIMULUS WORD WHISTLE.

	P_S	P_{M1}	P_{M2}
STOP	.1403	.1298	.0560
BLOW	.0864	.0555	.0600
TRAIN	.0678	.0882	.1060
SHRILL	.0518	.0307	.0300
NOISE	.0504	.0723	.0660
DANCE	.0472	-	-
MUSIC	.0428	.0079	.0110
SONG	.0403	.0327	.0370
SING	.0396	.0614	.0540
SOFT	.0325	-	.0010
SHOUT	.0307	.0059	.0030
BIRD	.0304	.0039	.0070
WIND	.0279	.0009	.0020
TUNE	.0269	.0485	.0320
LOUD	.0234	.0168	.0400
GO	.0233	-	-*
HAPPY	.0215	.0049	.0090
PIERCING	.0197	-	.0030
SOUND	.0189	.0386	.0610
BOY	.0172	.0257	.0370
WOLF	.0172	.0178	.0230
GAME	.0172	.0009	-*
CHOIR	.0144	-	-
NOSE	.0122	-	-*
DOG	.0090	.0366	.0180
PIANO	.0072	-	-
WATER	.0059	-	-*
VOICE	.0054	-	-*
RED	.0054	-	-*
BARK	.0052	.0009	-*
RECORD	.0049	-	-
GIRL	.0046	.0386	.0430
TRAFFIC	.0044	-	-
QUIET	.0044	-	-*
EAR	.0034	.0009	-*
ANIMAL	.0034	-	-
GAY	.0032	-	-*
MOUTH	.0032	.0148	.0180
FOOD	.0029	-	-
TALK	.0029	.0039	.0020
LOW	.0029	.0009	-*
SHORT	.0017	.0009	-*
PIPE	.0016	.0019	-
HIGH	.0011	.0059	.0070
SIGNAL	.0009	.0019	-*
SHARP	.0007	.0079	.0130
GONG	.0007	-	-
WOMAN	.0007	.0099	.0050
LAD	.0004	-	-*