

Mixing Binary and Continuous Connection Schemes for Knowledge Access

N.E. Sharkey, R.F.E. Sutcliffe and W.R. Wobcke*,
Centre for Cognitive Science,
University of Essex,
UNITED KINGDOM.

ABSTRACT

We present BACAS, a Binary and Continuous Activation System which is a parallel process content-addressable memory model. BACAS is designed for the representation and retrieval of 'knowledge of the world' for automatic natural language understanding. In its present form, BACAS is a two-layered system with 10 K-structures (like scripts) in the binary output macro-layer represented by 46 Threshold Knowledge Units and 184 processing elements (like action events) in the continuous activation micro-layer. We discuss the problems of combining two types of connection system and describe a simulation in which the system moves from one pattern to the next in response to external input. A new tool for connection systems, the pulse-out, is introduced. This is a device which replaces the Boltzmann Machine in creating energy leaps. The pulse-out also has the advantage, in the current system, of setting the state of the system a short Hamming distance from an appropriate pattern.

I INTRODUCTION

One of the central problems for any automatic natural language understanding device is that humans use language elliptically. People try to leave out much of the information they believe their readers/listeners share with them (Grice, 1975). Thus texts written by humans for humans are usually only a partial pattern of the information they are intended to convey. It must therefore be one of the first jobs of an automatic natural language understander to fill in the information missing from the input pattern. This technique has been employed for many schemata-driven story understanders (eg. Cullingford's (1978) SAM). We present a content-addressable memory retrieval mechanism which is designed to act as the core of a natural language understander. This differs from schema-appliers in that we use a distributed knowledge representation in a parallel process network.

The model we present here is a considerably extended and modified version of the Knowledge Access Network (KAN) (Sharkey, in press; Sharkey & Sharkey, in press). KAN combines the script concept (Schunk & Abelson, 1977) with a spreading activation parallel associative network account of memory. The development of this model has represented an attempt to capture the notion of stereotypical information without relying on 'packages' of

knowledge. KAN treats memory as continuous while behaving 'as if' it contained script-like entities. Furthermore, the KAN system differs from the Script Applier Mechanisms (eg. Cullingford, 1978) in the way it accesses knowledge. KAN does not need to search for a matching precondition to gain entry to its knowledge. Rather, the retrieval of KAN's knowledge is integrated with its understanding process. Also, unlike SAM, KAN can have more than one knowledge pattern active at the same time.

The new model BACAS (Binary And Continuous Activation System), is more distributed than KAN and utilizes both continuous and binary activation schemes. Like KAN, BACAS is architecturally very simple. It has two distinct domains of processing: a *micro-layer*, consisting of many processing elements which represent the system's conceptual knowledge, and a *macro-layer* consisting of a number of Threshold Knowledge Units (TKUs) which act as the only connecting stations between the micro-elements (see Figure 1). (A TKU, as its name suggests, is a variant of the Threshold Logic Unit (Minsky and Papert, 1969) and follows in the McCulloch & Pitts, 1943, tradition.) A collection of micro-elements linked to a TKU is called a *k-pattern*. Each processing element may belong to several k-patterns. The contribution of an element to a given k-pattern is represented by the weight of its connection to the TKU. This weight also tells us about the system's 'belief' about the likely relevance of an element to a TKU.

A TKU has both an activation level (continuous) and an output (binary). The continuous activation level is the communication between the macro-layer to the micro-layer. Evidence as to the relevance of a particular micro-element is collected by the TKUs and, once a TKU reaches threshold, its activation is broadcast to all the micro-elements connected to it, thus completing its k-pattern in the micro-layer. When a TKU is 'on' (defined later), its output is 1, otherwise it is 0. These outputs form the communication between TKUs. The macro-layer settles into a particular pattern, based on the evidence of the k-patterns which have been recognized.

In KAN, each TKU in itself corresponded to a script (Schunk and Abelson, 1977). We had therefore imposed the restriction that only one TKU could be on at a time. However, because of the psychologically observed 'handover period' (Sharkey & Mitchell, 1985), the system was designed to behave 'as if' two K-structures were simultaneously active. That is, in the transition period between successive K-structures, the micro-elements of more than one TKU could be active. For convenience, a simple conceptual device, a one-place F-register, was employed to shunt TKUs on and off. When a TKU was 'on' it was said to be in *focus*. In BACAS, like KAN, a

* We would like to thank the Economic and Social Research Council (Grant No. C 08 25 0015), the Science and Engineering Research Council and the Commonwealth Scholarship Commission in the United Kingdom for supporting this research.

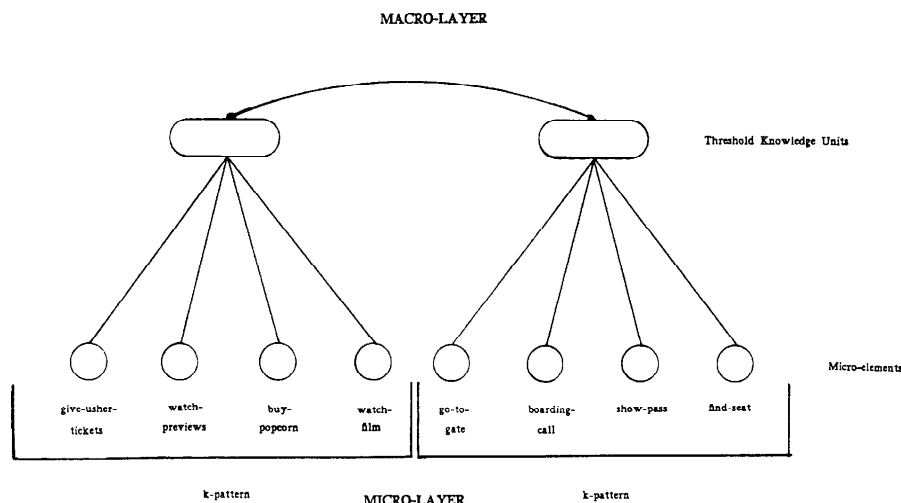


Figure 1: The k-pattern corresponding to the 'watch-film' scene of the 'going to the movies' K-structure, and a k-pattern corresponding to the 'boarding' scene of the 'catching-a-plane' K-structure. These TKUs are negatively connected by symmetrical weights. Two k-patterns and two TKUs are shown.

TKU in the 'on' state keeps the k-pattern of elements associated with it in an active state until it is switched off.

In BACAS, each TKU and its associated k-pattern corresponds to a scene. The collection of TKUs corresponding approximately to a set of scenes occurring together in a script (Schank & Abelson, 1977) is called a *K-structure*. The macro-layer of BACAS is designed to represent one such K-structure at a time. A TKU may occur in many K-structures: thus K-structures are like MOPs (Schank, 1982). For example, the K-structure for 'going-to-the-movies' now has six k-patterns and corresponding TKUs (choose film, stand-in-line, buy-ticket, enter-cinema, watch-film, exit-cinema). Our motivation for this change was to give BACAS more flexibility and to create the potential for dynamic modification of memory. In order for a BACAS K-structure to be in focus we now need several TKUs to be 'on' at once. This meant that we could no longer use the F-register. Instead, we have had to make the macro-layer strongly interconnected, that is, connect each TKU to every other TKU. In the most recent simulation we have used 10 K-structures and 46 TKUs. These are based on a set of empirically collected norms. (Galambos, 1982).

The problems that BACAS is designed to solve are:

1. How to ensure completion of K-structures in the macro-layer such that the appropriate k-patterns in the micro-layer would be completed.
2. How to move from one K-structure to another, that is, how to switch on the appropriate TKUs and switch off the inappropriate ones on the basis of partial information (eg. cashing a check and then going to the movies).

We turn now to examine each of the processing layers. We then give an example of the behavior of BACAS, showing how the above situations are handled, and finally discuss the choice of weights for the macro-layer.

II THE MICRO-LAYER

Processing in the micro-layer of BACAS (and KAN) is driven by an activation mechanism which is a variant of that of McClelland and Rumelhart (1981). Their threshold rule holds invariably for the micro-elements, but a TKU may be in one of three states, and its behavior varies accordingly:

1. A TKU in the *off* state accumulates activation from the micro-layer but it does not transmit activation to the micro-layer.
2. A TKU in the *on* state transmits a constant source of activation to the micro-layer but it does not receive activation from the micro-layer. This constant activation is the TKU's saturation value.
3. A TKU is in the *intermediate* state between the 'on' and 'off' states while it decays from saturation to resting. In the 'intermediate' state, a TKU continues to transmit its decaying activation until it reaches resting level, but it receives no activation from the micro-layer.

The dynamic behavior of a TKU is as follows. When a TKU in an 'off' state reaches threshold after accumulating activation from the micro-layer, it immediately assumes the 'on' state and its activation is set to saturation. It will eventually be set to the intermediate state from the macro-layer (see the next section). If the TKU decays to resting before being switched on from the macro-layer, it assumes the 'off' state again. It is only then that it will again receive activation from the micro-layer.

The idea behind 3, the intermediate state, was directly imported from KAN and is an important property of the model. The decaying activation on a TKU allows the 'handover period' mentioned above (Sharkey & Mitchell, 1985) so that there is a smooth transition between K-structures.

We can formally describe the micro-layer as follows. The net input $n_j(t)$ of activation to the j^{th} element at time t is the weighted sum of the activation passed to it by its neighbors thus:

$$n_j(t) = \begin{cases} 0, & \text{for a TKU which is not off,} \\ \sum_i a_i(t)w_{ij}, & \text{otherwise.} \end{cases}$$

where a_i is the activation on the i^{th} micro-element or TKU and w_{ij} is the strength of association from i to j . Note that in BACAS, the neighbors of a micro-element are just the TKUs connected to it, and the neighbors of a TKU are just the elements of its k-pattern.

It is assumed that each micro-element and TKU saturates at below a point S (we set $S = 1$). The effect $\xi_j(t)$ of activation input to a micro-element or TKU j at time t is defined by:

$$\xi_j(t) = n_j(t)(S - a_j(t))$$

Activation on an element decays over time at a rate proportional to the amount of activation above resting level. The activation of the j^{th} element at time $t+1$ (i.e., one time cycle after t) is given by:

$$a_j(t+1) = \begin{cases} S, & \text{for a TKU which is on,} \\ a_j(t) - \delta(a_j(t) - r_j) + \xi_j(t), & \text{otherwise.} \end{cases}$$

where r_j is the resting or base level activation of element j (which is positive), and δ is the constant of decay.

III THE MACRO-LAYER

Communication in the macro-layer is by means of the binary outputs of the TKUs and a *pulse-out* mechanism. The pattern completion method in BACAS is similar to that of the asynchronous, deterministic machine of (Hopfield, 1982). That is, from an 'initial' pattern, the machine will settle in a state which is a local energy minimum, where the energy of a state is defined by:

$$E = -\frac{1}{2} \sum_{ij} s_i w_{ij} s_j - \sum_i \mu_i s_i$$

where w_{ij} is the weight from the i^{th} to the j^{th} TKU and μ_i is the activation on the i^{th} TKU. Thus the activation of a TKU acts like external input in the formula of Hinton & Sejnowski (1983)*. To make a transition in the macro-layer, a unit j is chosen at random, and the change in energy ΔE_j is calculated:

$$\Delta E_j = \sum_i s_i w_{ij} + \mu_j$$

The state of the unit, s_j is then set as follows:

* Note that, because of the μ term, the energy landscape changes while the pattern completion mechanism is in progress, unlike Hinton & Sejnowski's method, where, because the external input is fixed during pattern completion, the energy landscape is also fixed.

$$s_j(t+1) = \begin{cases} \text{on,} & \text{if } \Delta E_j \geq 0, \\ \text{intermediate,} & \text{if } \Delta E_j < 0. \end{cases}$$

As with a TKU reaching threshold after accumulating activation from the micro-layer, a TKU switched on from the macro-layer has its activation set to its saturation value.

One problem which confronted us was that a language understander may need to apply a succession of different K-structures as it reads through a text. How do we get the system to move from one K-structure to the next? As a first attempt, we tried using a Boltzmann Machine (Hinton & Sejnowski, 1983). We treated each K-structure as a global energy minimum to be found using an annealing schedule. Following Hinton & Sejnowski, the TKUs which had fired were treated as fixed or *locked* units. However, this still left the problem of getting the system to move from one K-structure to the next. We viewed the target K-structure as a required global minimum and the current K-structure as an unwanted local minimum. The new K-structure could then have been found by initiating a new annealing schedule at the appropriate time. However to make this work, we would have had to have found a way of telling the system *when* to unlock the units of the current K-structure and *when* to begin a new annealing schedule. One possibility would have been to run the annealing schedule each time a new TKU fired. But it is logically unnecessary to do this, since not all of the TKUs which fire will signify a new K-structure.

Instead, we solved the problem by using a *pulse-out* mechanism to set up a new 'initial' pattern, close to the desired local minimum. The local minimum can then be found deterministically. The pulse-out works as follows. When a new TKU reaches threshold after accumulating activation from the micro-layer, it transmits a pulse. Any negatively connected 'on' TKUs are pulsed out, so that they will assume the intermediate state. As a result, when the new TKU belongs to a new K-structure, the bit string representing the state of the system is set a shorter Hamming distance away from the the new K-structure than any of the other K-structures. The pulse-out, in effect, creates a leap to a higher energy state (if the new TKU does not occur in any K-structures with currently active TKUs, the energy of the new state will be 0).

By using the pulse-out, we have avoided using locked units and an annealing schedule in conjunction with a Boltzmann Machine and, as a result, our system is deterministic.

IV PROCESS AND COMPLETION

In the BACAS model, when script information is presented sequentially to the system (eg. 'Sam went into a restaurant. He ordered a meal. He paid the bill.'), it activates a set of processing elements connected to a number of TKUs. For example the micro-element corresponding to 'he paid the bill' would be linked to TKUs associated with knowledge about restaurants, shops, garages, etc. (see Figure 2). The active elements broadcast activation to their TKUs until the activation on a TKU passes a preset threshold. When this happens the thresholded TKU is switched to the 'on' state. The 'on' TKU then transmits a constant activation source to the

active and inactive micro-elements of its associated k-pattern in the micro-layer. Thus the k-pattern most relevant to the current input is completed. This enables BACAS to draw inferences about the likelihood of occurrence of other events than those which occurred. When the activation in a k-pattern 'relaxes', the activation on the micro-elements of a k-pattern are said to represent their 'belief' as to their relevance to the input.

While the units in the k-pattern of the TKU that reached threshold are being activated, the macro-layer of BACAS settles into a local energy minimum which is the K-structure that BACAS 'believes' to be the most relevant to the input. The k-patterns of the TKUs in this K-structure are activated, thus completing a pattern of micro-elements which corresponds to a 1977 script. In this example, the 'restaurant' K-structure is completed.

When a TKU not in the current K-structure reaches threshold, it pulses out the TKUs that do not share a K-structure with it. Continuing with our example, should 'Sam went to the cinema and looked at the program' occur, the 'choose-film' TKU would fire. The pulse-out sets each of the TKUs in the 'restaurant' K-structure to the intermediate state. The TKUs and the corresponding micro-elements in their k-patterns then decay. Meanwhile, the macro-layer settles into the 'going-to-the-movies' K-structure, and activation begins spreading to the micro-elements of the other k-patterns in this K-structure, for example, 'buy-ticket' and 'enter-cinema'. Evidence for these TKUs is also collected from the micro-layer.

So far we have implemented the macro- and micro- layers separately, and the macro/micro interface has also been simulated. It now remains for us to put the two layers together.

V CHOOSING WEIGHTS

In this section, we discuss our procedure for choosing the weights on connections between the TKUs in the macro-layer. For BACAS the weights should ensure, amongst other things, the following two important properties. First, TKUs that never occur together in any K-structure should be negatively connected to guarantee the effective operation of the pulse-out mechanism. These negatively connected TKUs were all connected with a weight of -1. Second, each of the K-structures should be a stable state of the system. To ensure this, we set the connection between each pair of distinct TKUs which occur together in some K-structure to a small positive weight, c , and the weight from a unit to itself is set to 0. In the remainder of this section, we discuss how to choose c so that the K-structures are local minima.

To ensure that a given K-structure is a local minimum, we simply make its energy value lower than that of all the patterns which are a Hamming distance of 1 away from it. These patterns are its *neighbors*. That is, two states are neighboring if the bit strings representing the states differ in only one bit, eg. 1101010 is a neighbor of 0101010. Thus, we take a K-structure of n TKUs (ie, the n 'on' TKUs that make up the K-structure), and derive constraints on c by considering the change in energy between that K-structure and its neighbors. The change in energy, ΔE_j , created by altering the j^{th} unit in

a pattern from 1 to 0, is given by:

$$\Delta E_j = \sum_i s_i w_{ij} + \mu_j$$

where w_{ij} is either c , -1 or 0, and s_i is the output from the i^{th} TKU.

We first discuss the case where the j^{th} unit of a pattern changes from 1 to 0 (*downward neighbor*), and then examine its opposite, where the j^{th} unit changes from 0 to 1 (*upward neighbor*). First, when a TKU j changes from 1 to 0, (ie, the state changes from a K-structure with TKU j 'on' to one of its neighboring patterns), the difference in energy is $(n-1)c + \mu_j$. This is because there are $n-1$ units in the K-structure which are each connected to TKU j with positive weight c . This change in energy is always positive and so the K-structure has a lower energy than any of its downward neighbors.

Second, to ensure that a K-structure has a lower energy than any of its upward neighbors, we examine the difference in energy resulting from changing the j^{th} TKU from 0 to 1. Since this new TKU is not in the K-structure, it will be negatively connected to at least one of the TKUs in the K-structure*. If the new TKU is negatively connected to i TKUs in the K-structure, the change in energy is $(n-i)c - i + \mu_j$, since i TKUs have weight -1 to the new TKU while the other $n-i$ TKUs in the K-structure have weight c . If we choose a c to ensure that this quantity is always negative, the K-structure will have a lower energy than all of its upward neighbors. We can only do this if $i > 1$. If this holds, the maximum value that the change in energy can take is when $i = 2$ and the activation of the new TKU is at saturation. (ie, when each of $n-2$ TKUs in the chosen K-structure is contained in K-structures which also contains the new TKU). This maximum value is $(n-2)c - 1$, since the saturation value of a TKU is 1. To make the maximum negative, we choose a c so that $(N-2)c < 1$, where N is an upper bound on the number of TKUs a K-structure may possess. Thus, with this choice of c , we have guaranteed that K-structures are local minima provided that if we choose any K-structure and any new TKU, $i < 1$.

A further desirable property of BACAS is that, at all times, the system should have an hypothesis as to which K-structure is relevant. Thus with an ambiguous input text, the system would fix on just one of the interpretations of the input. The K-structure selected by the system as its hypothesis might, however, be dependent on the order of firing of the TKUs. For example, in our system, on being presented with 'stand-in-line' and 'buy-ticket', we want the machine to settle into either the 'going-to-the-movies', 'catching-a-train' or 'taking-the-subway' K-structure, rather than some other pattern.

The main obstacle to BACAS's achieving a single hypothesis with ambiguous input is that the union of two K-structures could be a local minimum. The system could settle into such a state when a TKU from a new K-structure comes on and the pulse-out leaves the machine in a state which is a shorter Hamming distance from the union of the new and the old K-structures than from either of the K-structures. We avoid this by choosing a c so that the union of any two K-structures has a higher energy than any of its downward neighbors. Since our

pattern completion mechanism only makes moves to lower energy states, we can guarantee that the union of two K-structures can never be reached from below. The calculation is as follows. Suppose the two patterns contain m and n TKUs. Let the union of the K-structures and its downward neighbor differ in TKU_j , which is a member of the first K-structure. If i is the number of TKUs in the second K-structure that do not share K-structures with TKU_j (ie, those which are negatively connected to TKU_j), the change in energy from the K-structure to the neighboring pattern is $(m-1+n-i)c - i + \mu_j$. An upper bound on this quantity (where $i > 1$) is $(m-1+n-2)c - 1$. Thus, the change in energy is always negative if we choose c such that $(N-1)c < 0.5$, where N , as above, is an upper bound on the number of TKUs in a K-structure.

In consequence, when a shared TKU is the first to reach threshold, the macro-layer settles into one of the K-structures containing the shared TKU. So even if it turns out to be the wrong K-structure, a small amount of activation will have flowed to the elements in its k-patterns.

If the very unusual situation occurred where two TKUs from different K-structures thresholded simultaneously, the system would settle into one of the K-structures, thus resolving this 'contradiction' by ignoring some of the evidence it collects.

The 10 K-structures we have used to test BACAS are based on empirically collected norms, (Galambos, 1982). For these K-structures, only two TKUs, 'stand-in-line' and 'buy-ticket', are shared. The maximum number, N , of TKUs per K-structure is 6. Thus choosing c to be 0.2, so that $(N-2)c < 1$, results in the K-structures being local minima with the desired behavior.

VI CONCLUSION

Our aim here has been to present a content-addressable memory retrieval mechanism for using 'knowledge of the world' in a natural language understander. Simplicity was our goal and we have left out a lot of what is important to language understanding. There is no representation of causality in BACAS, nor a mechanism for the temporal ordering of units, nor for role binding. Although these are obvious necessities, we are trying to get the retrieval mechanism correct before tackling these problems.

We have demonstrated the usefulness of combining two types of connection system and we have described a technique to enable a connection system to move from one pattern to the next in response to external input. Our pulse-out mechanism replaces the Boltzmann Machine in creating energy leaps without the need for locked input. Our current research is geared towards both developing a learning mechanism for BACAS and building the natural language front-end.

* A K-structure cannot be a subset of another K-structure.

REFERENCES

- [1] Cullingford, R.E. (1978) 'Script Application: Computer Understanding of Newspaper Stories.' Technical Report 116, Department of Computer Science, Yale University. Ph.D. dissertation.

- [2] Galambos, J.A. (1982) 'Normative studies of six characteristics of our knowledge of common activities.' Cognitive Science Technical Report 14, Yale University.
- [3] Grice, H.P. (1975) 'Logic and conversation.' In P. Cole & J.L. Morgan (Eds) *Syntax and Semantics*. (Vol 3): *Speech acts*. New York: Seminar Press.
- [4] Hinton, G.E. & Sejnowski, T.J. (1983) 'Analyzing cooperative computation.' *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*. Rochester, NY, May.
- [5] Hopfield, J.J. (1982) 'Neural Networks and Physical Systems with Emergent Collective Computational Abilities.' *Proceedings of the National Academy of Sciences, U.S.A.*, 79, 2554-2558.
- [6] McClelland, J.L. & Rumelhart, D.E. (1981) 'An interactive model of context effects in letter perception: Part 1. An account of basic findings.' *Psychological Review*, 88, 375-407.
- [7] McClelland, J.L. & Rumelhart, D.E. (1985) 'Distributed memory and the representation of general and specific information.' *Journal of Experimental Psychology: General*, 114, 159-188.
- [8] McCulloch, W.S. & Pitts, W.H. (1943) 'A logical calculus of ideas imminent in nervous activity.' *Bulletin of Mathematical Biophysics*, 5, 115-133.
- [9] Minsky, M. & Papert, S. (1969) *Perceptrons*. Cambridge, Mass: MIT Press.
- [10] Schank, R.C. & Abelson, R.P. (1977) *Scripts, Plans, Goals and Understanding*. New Jersey: Lawrence Erlbaum.
- [11] Schank, R.C. (1982) *Dynamic Memory*. Cambridge: Cambridge University Press.
- [12] Sharkey, N.E. & Mitchell, D.C. (1985) 'Word recognition in a Functional Context: The Use of Scripts in Reading.' *Journal of Memory and Language*, 24, 253-270.
- [13] Sharkey, N.E. (1986) 'A Model of knowledge-based expectations in text comprehension.' In J.A. Galambos, J.B. Black and R.P. Abelson (Eds) *Knowledge Structures*. New Jersey: Lawrence Erlbaum.
- [14] Sharkey, N.E. & Sharkey, A.J.C. (1986) 'KAN: A knowledge access network model.' In R. Reilly (Ed) *Dialogue Failure*. Amsterdam: Elsevier, North Holland.