Proceedings of the third annual conference of the Cognitive Science Society.
Berkeley, California, August 19-21, 1981.

Retrieving General and Specific Information from Stored Knowledge of Specifics

James L. McClelland
University of California, San Diego

We often attribute the human ability to generalize from past experience to the use of stored representations (schemas, prototypes, etc.) in which generalizations are explicitly represented. This view is very appealing, but it raises two problems. First, there needs to be some mechanism for arriving at generalizations that are not stored explicitly, since it is unlikely that memory contains explicit representations that anticipate all of the possible generalizations we might ever wish to make. Second, we must explain how generalizations which are stored explicitly were obtained in the first place.

A mechanism that could induce generalizations from stored representations of specific objects or events could solve both these problems at once. It would explain how we could generalize when no explicit generalization is stored, and it would also suggest how we might have induced those generalizations which are stored.

Such a mechanism would also force us to consider whether we really store generalizations explicitly at all. If we can generate generalizations from stored representations of specific objects when we need to, explicit representation of these generalizations might turn out to be unnecessary.

Medin and Shaffer (1978) have suggested a first step toward the kind of mechanism I have in mind. Their model explains how we can assign a category label to a new object, based only on stored knowledge of the properties of previously encountered objects and the category labels that have been assigned to them. Their ideas can be extended to suggest how we may be able to do such things as answer questions about the general characteristics of classes of objects we have experienced before, and to fill in plausible default values for unspecified attributes of new exemplars.

The basic idea is that representations of previously-experienced exemplars stored in memory are activated via a spreading activation mechanism. Activated exemplars themselves activate representations of their properties. Mutually exclusive property values compete so that properties which are supported by a large subset of the active instances of the category are reinforced and become strongly active while those which are not are suppressed. Such a mechanism has recently been proposed by Glushko (1979) to account for our ability to construct apparently rule-guided pronunciations of nonwords (e.g., MAVE) without actually having any rules, and has been used by Rumelhart and me (McClelland and Rumelhart, in press; Rumelhart and McClelland, in press) to account for facilitation of perception of letters in words and nonwords. In both of these applications, the activation/competition mechanism is used to generate apparently rule-governed performance from stored knowledge of specific words.

I will illustrate the mechanism I am proposing by showing how it can be used to generalize from stored representations of specific objects. The representations of the objects are highly simplified, and are not sufficient to capture the varieties of structure of real objects. It is not my intention to advocate the representation. Rather, I use it to explicate the generalization mechanism, which is the main focus of interest here. We shall see that, even with a simplified representational system, the activation and competition mechanism can construct the general properties of classes of objects from stored knowledge of exemplars. It can also generalize along an indefinite number of different lines, retrieve the specific characteristics of particular exemplars, and fill in plausible default values for missing properties.

Table 1
The Jets and The Sharks

Name	Gang	Age	Edu	Mar	Occupation
Art	Jets	40's	J.H.	sing.	pusher
Al	Jets	30's	J.H.	mar.	burglar
Sam	Jets	20's	COL.	sing.	bookie
Clyde	Jets	40's	J.H.	sing.	bookie
Mike	Jets	30's	J.H.	sing.	bookie
Jim	Jets	20's	J.H.	div.	burglar
Greg	Jets	20's	H.S.	mar.	pusher
John	Jets	20's	J.H.	mar.	burglar
Doug	Jets	30's	H.S.	sing.	bookie
Lance	Jets	20's	J.H.	mar.	burglar
George	Jets	20's	J.H.	div.	burglar
Pete	Jets	20's	H.S.	sing.	bookie
Fred	Jets	20's	H.S.	sing.	pusher
Gene	Jets	20's	COL.	sing.	pusher
Ralph	Jets	30's	J.H.	sing.	pusher
Phil	Sharks	30's	COL.	mar.	pusher
Ike	Sharks	30's	J.H.	sing.	bookie
Nick	Sharks	30's	H.S.	sing.	pusher
Don	Sharks	30's	COL.	mar.	burglar
Ned	Sharks	30's	COL.	mar.	bookie
Karl	Sharks	40's	H.S.	mar.	bookie
Ken	Sharks	20's	H.S.	sing.	burglar
Earl	Sharks	40's	H.S.	mar.	burglar
Rick	Sharks	30's	H.S.	div.	burglar
Ol	Sharks	30's	COL.	mar.	pusher
Neal	Sharks	30's	H.S.	sing.	bookie
Dave	Sharks	30's	H.S.	div.	pusher

I will illustrate the features of the model by considering how it can be used to retrieve information about the members of two gangs called the Jets and the Sharks. Characteristics of hypothetical members of these two gangs are listed in the Table 1.

The model's knowledge of these individuals is captured in a node network. Each node is a simple processing device which accumulates excitatory and inhibitory inputs from other nodes continuously and adjusts its (real-valued) output to other nodes continuously in response, much as a neuron adjusts its rate of firing in response to a varying pattern of excitatory and inhibitory inputs.

The model has a node for each of the individuals it knows and a node for each of the properties or attributes these individuals may have. The former are called instance nodes and the latter are called property nodes. There is a property node for each individual's name, one for each gang, one for each age range, one for each educational level, and so on. Property nodes are arranged into groups or cohorts of mutually exclusive values. The instance nodes are also treated as a cohort of mutually exclusive nodes. In the following Figure, the instance nodes have been placed in the center with the property nodes all around. Nodes within a cohort (bounded region) are mutually inhibitory.

The system's knowledge of an individual consists simply of an instance node and a set of bi-directional excitatory links between it and the nodes for the properties that individual is known to have. For example, the system's representation of Lance is an instance node with mutual excitatory connections to the name node "Lance", the gang membership node "Jet", the age node "20's", the education node "Junior High", the marital status node "married", and the occupation node "burglar".

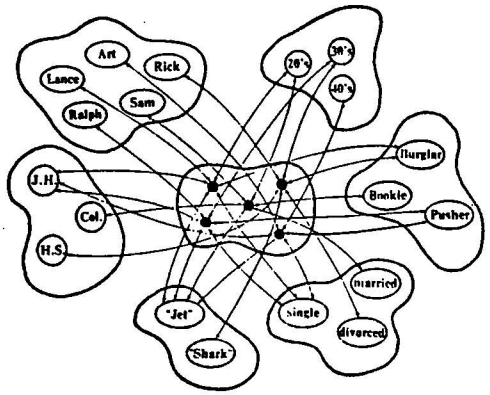

Figure 1. The representation of several of the individuals listed in Table 1.

Filling in Properties in Response to a Probe

The system is queried by presenting it with a probe. For example, to find out about the properties of the Jets we can probe the system by activating the property node "Member of Jets". A probe might be a name or any other single property, or it may consist of a list of properties.

Before a probe is presented, each node is assumed to be at rest, with an activation value below 0. Probe presentation causes an excitatory input to be applied to each node specified in the probe. This excitatory input is allowed to stay on, and as time passes it drives the activations of the specified property nodes above 0, into what is called the active range. Active nodes send excitatory signals to the instance nodes they are linked to and send inhibitory signals to the other nodes in the same cohort. These signals are graded, and their strength is proportional to the source node's activation. As processing continues, some of the instance nodes become active. They then begin to excite the property nodes they are connected to, and to inhibit all the other instance nodes. Eventually, property nodes not present in the probe may become activated.

The excitation and inhibition processes are allowed to go to equilibrium. At this point, the system has generally activated property nodes for properties not specified in the probe. These activations are the system's response to the probe. If all of the active instance nodes "agree" on a property, the node for that property will tend to be strongly activated. On the other hand, if they all specify different values within the same cohort, many values will become partially activated and they will all tend to cancel each other out. In any case, what is filled in can then be used as a basis for overt response to the probe. For example, a statement of the typical age of the members of the Jets could be based on the resulting pattern of activation over the age nodes. I will go through some examples of what the system fills in in response to various probes after giving a few more details of the working of the model.

Quantitative Details

The net input to node i at time t is given by:

$$input_i(t) = p_i(t) + E\sum_j e_{ij}(t) - I\sum_j i_{ij}(t).$$

$p_i(t)$ stands for the probe input to node i. It is set to +.2 if the probe drives node i and to 0 otherwise. The $e_{ij}(t)$ are the activations of the active excitors of node i and the $i_{ij}(t)$ are the activations of the active inhibitors of node i. The constants E and I are simply weights which modulate the excitatory and inhibitory effects of the input. Their values (.05 and .03) are the same for all nodes except as noted below.

The effect of the net input to node i is modulated by the current activation ($a_i(t)$). If the net input is excitatory (i.e., greater than or equal to 0). then the effect is

$$effect_i(t) = (M-a_i(t))input_i(t)$$

If the net input is inhibitory (i.e., less than 0) then the effect is

$$effect_i(t) = (a_i(t)-m)input_i(t)$$

Here M stands for the maximum possible activation of the node and m stands for the minimum. This formulation ensures that the activation of each node stays between the maximum and minimum values, which are set to 1.0 and -.2 respectively.

There is a tendency for the activation of each node to decay at some rate D back to its resting value R. This tendency is subtracted from the effect of the net input to the node to determine the rate of its activation:

$$d(a_i(t))/dt = effect_i(t) - D(a_i(t)-R) .$$

The values of D and R are .05 and .1.

Simulation

The behavior of the system described above is simulated on a digital computer by using discrete rather than continuous time. One every tick of the discrete clock, the activations of each node are adjusted to reflect the effects of the activations of other nodes at the end of the previous tick. The time slices are kept thin by using small values for E, I, and D, so that the approximation to a continuous system is quite close.

Examples of the Model's Behavior

Let us examine the system's response to the probe "Member of the Jets". Presentation of the probe causes the "Jet" node to become active, and this in turn sends activation to the instance nodes of all of the members of the Jets. As they become active they send excitation to the nodes for their properties. These nodes in turn reinforce the activations of those jets with active properties. After about 200 cycles the pattern of activation over the property nodes has stabilized at the following values:

Name:	—			
Gang:	Jets	.869		
Age:	20's	.663		
Ed:	J.H.	.663		
Mar:	Sing.	.663		
Occ:	Pusher	.334	Bookie	.334
	Burglar	.334		

Activations for instance nodes are omitted to save space. All property nodes not mentioned are below zero activation. Based on these activations the model could generate a list of its conception of the typical properties of the Jets. In the case where only one possibility is active, the system would simply report that value. Where multiple possibilities are active, it could either list the set of possibilities or make a probabilistic choice from among the alternatives.

In this case the active age, education, and marital status properties are the ones which are typical of the Jets. Though no Jet has all three of these properties, 9 out of 15 of the Jets are in their 20's, 9 have only Junior High educations, and 9 are single. The occupations are divided evenly among the three possibilities. Thus, the model tends to activate the node on each dimension which is most typical of the members of the gang, even though it has never encountered a single instance with all of these properties, and has no explicit representation that the Jets tend to have these properties.

An interesting feature of the model is that it can retrieve the typical properties of any subset of individuals matching an arbitrary conjunction of specifiable properties. For example, we can probe with the properties "Age in 20's" and "Junior High Education". Four individuals have these two properties. All of them are Jets and Burglars by trade. Two of them are married and two divorced. The response of the system reflects these facts:

Name:	Lance	.127	John	.127
	Jim	.094	George	.094
Gang:	Jets	.732		
Age:	20's	.855		
Ed:	J.H.	.862		
Mar:	Mar.	.589	Div.	.389
Occ:	Burglar	.721		

In this case the instance nodes for the four individuals matching the probe become strongly enough activated to drive the activations of the corresponding name nodes above threshold. Lance and John get more active than Jim and George because the instance node for A1, a married individual who is very similar to Lance et al., becomes slightly activated, thereby boosting the activation of the "married" node and causing Lance and John to gain a slight edge.

The model can also be used to retrieve the properties of a particular individual. In so doing, it exhibits the tendency to fill in "default" values for unknown properties of an instance. To illustrate this, we can delete the link between the instance node for Lance and the "burglar" node and then see what happens when we present the name "Lance" as a probe. The Lance name node becomes active and excites the corresponding instance node. This excites the nodes for the known properties of Lance. These then excite the nodes for other individuals who share these properties. Finally, they in turn excite the nodes for properties that they share. When the pattern of activity finally stabilizes (in about 400 cycles) the model has filled in an occupation for Lance.

Name:	Lance	.799		
Gang:	Jets	.710		
Age:	20's	.667		
Ed:	J.H.	.704		
Mar:	Mar.	.552	Div.	.347
Occ:	Burglar	.641		

The value filled in is shared by the other individuals who are most similar to Lance (namely John, Jim and George). At equilibrium the different marital situations of these individuals are also reflected in the pattern of activation. The model has blended its representation of Lance with its representation of other very similar instances.

This kind of blending can be a good or a bad thing, of course. It is sometimes important to know what we really know about something rather than what we might plausibly assume based on our knowledge of similar things. Fortunately, a single parameter of the model -- the strength of mutual competition among instances -- determines whether the model will tend to fill in values from partial activations of related instances. If active instances inhibit each other strongly, then the most strongly activated instance will tend to dominate the pattern of activation and keep other instances from "contaminating" the information retrieved. In the Lance example, if the strength of instance-to-instance inhibition is increased from .03 to .05, the instance node for Lance dominates the instance nodes and the others are kept from getting active so they cannot activate the missing occupation or the competing marital status. Thus, the model can either retrieve what is actually known about an instance or it can fill in missing properties from the common properties of similar instances.

In summary, the model I have described is capable of generalizing along a number of different lines about the shared properties of specified subsets of familiar objects. It can also retrieve what it knows about specific instances, and, if desired, fill in plausible default values for unknown properties of the retrieved individuals. It can induce generalizations as it needs them across novel partitions of the knowledge base. Since these are many of the behaviors which have led workers in various fields of cognitive science to assume we explicitly store generalizations, the model raises the possibility that this assumption, however plausible, may not necessarily be true in all cases.

There are many more steps to be taken, of course. For one thing, the model needs a representational system which can capture more highly structured knowledge. How the model can be extended in this way while preserving its interesting properties is currently being explored.

Acknowledgements

The work reported here was supported by NSF Grant BNS79-24062. I would like to thank Steve Draper, George Mandler, and Don Norman for very useful comments.

References

Glushko, R. J. The Organization and activation of orthographic knowledge in reading words aloud. Journal of Experimental Psychology: Human Perception and Performance, 1979, 5, 674-691.

McClelland, J. L., & Rumelhart, D. E. An interactive activation model of the effect of context in perception, part I. Chip Report 91, Center for Human Information Processing, University of California, San Diego. La Jolla, California, 1980.

Medin, D. L. & Shaffer, M. M. Context theory of classification learning. Psychological Review, 1978, 85, 207-235.

Rumelhart, D. E., & McClelland, J. L. An interactive activation model of the effect of context in perception, part II. Chip Report 95, Center for Human Information Processing, University of California, San Diego. La Jolla, California, 1980.