THREE SHORT PAPERS ON LANGUAGE AND CONNECTIONISM

Technical Report AIP-1

J. L. McClelland and Mark St. John

Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213 U.S.A.

29 September 1987

This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Number N00014-86-K-0678, and by ONR Contract Number N00014-82-C-0374. Reproduction in whole or in part is permitted for purposes of the United States Government. Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 1	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Carnegie-Mellon University 6b. OFFICE SYMBOL (*If applicable*)	7a. NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research (Code 1133)
6c. ADDRESS (*City, State, and ZIP Code*) Department of Psychology Pittsburgh, Pennsylvania 15213	7b. ADDRESS (*City, State, and ZIP Code*) 800 N. Quincy Street Arlington, Virginia 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization 8b. OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0678
8c. ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS p400005ub201/7-4-86 PROGRAM ELEMENT NO: N/A; PROJECT NO: N/A; TASK NO: N/A; WORK UNIT ACCESSION NO: N/A
11 TITLE (*Include Security Classification*) Three Short Papers on Language and Connectionism	
12 PERSONAL AUTHOR(S) J. L. McClelland and M. St.John	
13a. TYPE OF REPORT Technical 13b TIME COVERED FROM 86Sept15 TO 91Sept14	14 DATE OF REPORT (*Year, Month, Day*) 1987 September 29 15. PAGE COUNT 19
16 SUPPLEMENTARY NOTATION	

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
Cognitive Psychology, Learning, Language, Connectionism, PDP, PP attachment, Case Assignment, Verb tense

19 ABSTRACT (*Continue on reverse if necessary and identify by block number*)

This technical report contains three short articles on different aspects of language and connectionism. Together, the articles illustrate both the promise and the challenges facing the application of connectionist models to central issues in language processing. The first paper, (*Reconstructive memory for sentences*, by St. John and McClelland) describes a connectionist model in which background knowledge is used to aid recall and fill in missing arguments in sentences. The second, (*Parallel distributed processing and role assignment constraints*, by J. L. McClelland) discusses the application of connectionist models to the problem of using semantic/pragmatic constraints to processing sentences like "John ate the cake that his mother baked in the oven" as opposed to "John ate the cake that his mother baked in the dining room". The third paper gives a brief overview of the model of past tense learning developed by Rumelhart and McClelland.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Alan L. Meyrowitz	22b TELEPHONE (*Include Area Code*) (202) 696-4302 22c. OFFICE SYMBOL N00014

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

In press: Ohio University Inference Conference, 1986

Reconstructive Memory for Sentences: A PDP Approach

Mark F. St. John James L. McClelland

Department of Psychology
Carnegie-Mellon University

ABSTRACT

Two important principles help explain a number of memory phenomena: schematic reconstruction and distributed processing. Schematic reconstruction says that past knowledge is used to embellish and interpret a new encoding, and embellish and reconstruct old sentences during retrieval. A Parallel Distributed Processing mechanism is described that begins to embody this processing principle. The mechanism simulates 1) inferring elaborative propositions, such as instruments, 2) instantiating general nouns with more specific nouns related to the current context, and 3) integrating propositions from a number of sentences. The mechanism uses a distributed, feature representation of concepts that allows generalization, completion, and modification of the input to occur during initial processing. By implementing reconstructive processing in a distributed architecture, the mechanism is able to produce the memory phenomena introduced above.

INTRODUCTION

Given the large number of experiments on memory for sentences, it is important to develop an understanding of how their effects cohere. A good approach to this goal is to develop an underlying cognitive mechanism that can account for the varied effects. As a start, one can look for effects which seem to embody some underlying principle and then develop a mechanism that instantiates that principle.

One such principle is schema-based construction and reconstruction of propositions and word meanings. Bartlett in his famous War of the Ghosts experiment (1932) demonstrated how subjects' knowledge of stories interacted with a novel story to create a representation of the novel story. Bartlett's subjects modified the story to more closely resemble stories with which they were familiar. If they forgot aspects of the story, they filled them in with elements drawn from their schematic knowledge of stories, fitted to the context. This idea of selecting a schema and then using that schema to interpret and elaborate input can be applied to a wide range of sentence memory phenomena.

The implementation of schematic reconstruction involves the use of another principle: distributed processing. We have found that the characteristics of schematic reconstruction: contextually appropriate elaborations and inferences, and cross-sentence integration, are well handled by a distributed processor. When the representation and process is distributed across a large number of processors, these characteristics emerge naturally.

We have selected three effects involving a range of tasks which begin to cohere when seen from the vantage point of schematic reconstruction. In the course of this paper, we will interpret the phenomena from this perspective, describe our efforts toward creating a mechanism which embodies this principle, and then describe several simulations which begin to account for the experimental effects.

THREE PHENOMENA

Elaborative inferences. A large number of experiments have addressed the issue of what inferences are drawn during sentence processing. These propositional inferences can be divided into two classes: those that are required for comprehension and those that are merely elaborative. Required inferences must be drawn while the sentences are processed if the sentences are to be understood. Anaphoric referents, for example, must be drawn on-line. Knowing the instrument implied by a verb, however, often is not essential for a reasonable understanding of the sentence; rather, it merely serves to embellish the input. For example, in the sentence, "The sailor swept the floor," knowing whether or not he used a broom is largely unimportant for comprehension. Further, the number of elaborative inferences that might possibly be drawn from a sentence is unbounded. All manner of inferences about the agents, actions,

and context of a sentence can be drawn to embellish the input. The question many researchers have asked, therefore, is that given elaborative inferences are not required and infinite in number, what are the criteria for drawing them?

One experiment that has helped to illuminate this issue was conducted by McKoon and Ratcliff (1981). They presented subjects with a short paragraph, the final sentence of which implied a specific instrument. McKoon and Ratcliff then tested whether or not the instrument had been primed by the final sentence. They argued that priming of the instrument indicated that the instrumental inference had been drawn.

Interestingly, they found that the extent of the priming depended upon the strength of the relationship between the action described in the final sentence and the instrument. When the instrument was strongly related to the action, the inference was made, but when the instrument as only weakly related, the inference was not made. For example, hammer was inferred as the instrument of pounding a nail, but mallot was not.

This experiment does not tell us exactly what effect semantic relatedness has on making inferences. We believe that moderate relatedness between an action and an instrument should produce moderately active instruments. We understand inference making to consist of combining a large amount of evidence to produce more or less certain conclusions. Schematic factors, like how strongly related the use of a hammer is to the action of pounding a nail, affect the activation of conclusions, like that a hammer was actually used.

If inferences can be activated to a greater or lessor degree, then each argument of a proposition must have its own activation value. To produce the effect semantic relatedness has on the arguments' activation, an interaction must exist between the input and the long-term memory knowledge. Each propositional argument influences, and is influenced by, every other argument. This interaction occurs when the input activates an appropriate schema for the proposition. This schema then tries to embellish the input by priming additional propositional arguments. Arguments that are strongly implied are strongly activated, and arguments that are only weakly implied are only weakly activated.

Concept instantiation. Schematic reconstruction plays a prominent role in memory for individual words, as well. Anderson & Ortony (1975) and Anderson, Pichert, Goetz, Schallert, Stevens, and Trollip (1976) initiated a series of experiments demonstrating that subjects instantiate general nouns in sentences with more specific nouns related to the contexts of the sentences. It was shown, for example, that in the sentence, "The container held the apples," that the general noun "container" was instantiated with, or replaced by, the more specific noun "basket." In a recall test, a specific noun related to the context, such as "basket" proved to be a better cue than unrelated specific nouns or the general noun actually presented. Garnham (1979) demonstrated a similar effect for verbs.

These effects again demonstrate the ability of the sentence memory system to use long-term memory knowledge about propositions to interpret and embellish the input sentence. In this case, the propositional schema is used to embellish and modify individual words. Further experimentation, however, captured the criticism that it seems unlikely that the memory system would be able to select a particular specific noun in all but a few cases. Gumenik (1979) showed that a number of specific nouns related to the context, rather than just one, acted as good retrieval cues. A better explanation requires an appeal to the idea that concepts are defined by an underlying, distributed, feature-representation. The representation consists of a large number of semantic features whose activation specifies the strength of their presence. The pattern of activity over this set of features, therefore, dynamically determines what concepts are active in memory.

The schema activated by the input can modify the activation of the features representing a concept, just as it could modify the activation of an argument as a whole. When the system encounters a general noun, it retrieves the common features of nouns which fill its role in the proposition. By allowing each such feature to be modified individually, the instantiation effect can occur without requiring the selection, or substitution, of particular words. The reason the specific noun is a better recall cue follows from this modification process. During encoding, the representation of the general noun is modified sufficiently that the representation of the specific noun is actually closer to the result than the original representation of the general noun.

Similar to the instrument inference findings, these findings imply that the features of individual concepts are also active along a continuum, rather than simply on or off. This ability to partially activate features allows arbitary subtlety in the on-line definitions of words since the context can modify their features.

Proposition integration. The third experimental effect concerns the integration of propositions from a number of separate sentences. Bransford and Franks (1971) demonstrated that subjects can produce a holistic, schematic representation of a complex sentence from a number of simpler sentences. Bransford and Franks taught subjects a complex sentence such as "The ants ate the sweet jelly on the table in the kitchen," by asking them to study simpler sentences like, "The ants ate the jelly," and "The jelly was on the table in the kitchen." Subjects demonstrated their schematic knowledge of the complex sentence during a later recognition task. Subjects' confidence in recognition was based on the number of propositions in the test sentence. In fact, even when it was never studied, the sentence with the most propositions, the complete complex sentence, was recognized with the greatest confidence. Bransford and Franks concluded that the subjects had created an integrated representation of the entire set of studied sentences. The subjects then compared the test sentence with this representation and responded based on the similarity between the two.

Further experimentation (Reitman and Bower, 1973) showed that subjects can retain some information about specific items: confidence in recognizing new items was lower than in recognizing old items. Still, subjects recognized both old and new sentences with greater confidence as the number of propositions in the test sentences increased. From these studies, and others, it appears that building a schematic representation, and testing it against the input, is a ubiquitous process in sentence memory.

Implications for the model. Together these experiments provide a set of constraints on the mechanism for processing sentences. The Branford and Franks experiment shows that the mechanism must be capable of creating an integrated representation of a group of separately presented sentences. The resulting integrated structure must then be available to a comparison process to determine the similarity between itself and an input. The McKoon and Ratcliff study shows that this knowledge must be accessible by on-line processes and applicable to creating an elaborated representation of the input. Further, the mechanism must be able to add propositions to the input in such a way that the added propositions are activated to a greater or lesser extent depending on the characteristics of the input and long-term memory knowledge. The Anderson and Ortony experiment further shows that the propositional knowledge must also be available to the processing of individual words within the input to subtlely alter and instantiate word meanings to reflect long-term memory knowledge. Rather than making discrete choices between words, the process colors the meanings of words in appropriate ways.

These constraints can be satisfied if words and propositions is organized in a distributed manner in which words and propositions are represented as a large number of more or less active features. Particular words and propositions are represented by the pattern of activity across the set of features. The semantic distance between two concepts can be described as the difference between their activation patterns. Concepts that are very similar will have very similar activation patterns. If a group of patterns are similar enough, they can be labeled with a single word, and described as different senses of that word. Container, for instance, has a variety of meanings, and therefore, can label a number of similar activation patterns. If a pattern falls within the range of patterns labeled by a word, even though it may be a novel pattern, it receives the common label. The particular concept patterns established after a sentence has been processed depends upon the dynamic cooperation between the input and the long-term memory knowledge. It is likely that this process will create novel patterns on each occasion. It is exactly this ability of a word to label many related patterns that allows the system to combine particular instances and create schematic knowledge. It also allows the system to modify the definition of a concept without requiring a discrete choice between concepts and thereby allow "container" to mean different things in different contexts.

In addition to representation, the effects also imply a style of processing. Both bottom-up and top-down processing are strongly suggested by the empirical findings. In addition to the important role that the actual input plays bottom-up, schematic information works to modify and embellish the encoding top-down. Schematic information adds propositional arguments and modifies the encoding of existing arguments. The dynamic interaction of these two processing components creates the unique and contextually integrated encoding of each sentence.

THE MODEL

The approach we have chosen to use in modelling these effects is Parallel Distributed Processing (Rumelhart & McClelland, 1986). In this approach, computation is performed by a

large network of simple, interconnected processors. These processors send and collect activation via weighted links. While each processor acts only on information locally available to it, the entire network of processors is able to produce global solutions. Input to the network can be viewed as weak constraints on a solution. The processors interact to apply this large number of weak constraints in parallel. The weighted links determine how the processors will interact, and because these links can be modified by the processors, the network can learn to solve problems better over time.

PDP is well suited to the task because it is designed for distributed processing and naturally works to combine information encoded in the weights with the current input to embellish and reconstruct an input. PDP, therefore, embodies the style of processing implied by the experimental effects. Each argument of a proposition can be viewed as a weak constraint on a global solution for the encoding of a sentence. The arguments interact, drawing on past experience encoded in the weights, to produce a solution. A similar style of processing appears evident at the concept-feature level: the context provides a large number of weak constraints on the features of each word in the sentence, and these constraints similarly interact to arrive at a solution for the representation of a word.

Further, PDP networks are good at creating prototypes and schematic representations from a large number of instances, and then applying this information as new inputs are processed. This quality of the networks can be used to perform the argument and feature modifications found in the empirical results. The idea that schematic processing can be viewed as the interaction of each element in the schema is borrowed from Rumelhart, Smolensky, McClelland, and Hinton (1986).

Many assumptions must be made to create a working model. Some of these assumptions will be theoretically important, others merely convenient. The important assumptions in this model are as follows. The experimental effects presented above can be well explained by schematic reconstruction. Further, the experiments strongly suggest the nature of several aspects of sentence processing. Arguments are active to a greater or lesser degree in the propositional representation of the sentences. Their activation is mediated by the semantic relatedness between the action and the instrument, and more generally, between all elements of the proposition. Concepts are defined by underlying, distributed features which can also be modified by other elements in the proposition. And finally, confidence in recognition is influenced by the semantic similarity between the input and the long-term memory knowledge.

Representation. The model contains three levels of representation. There is a sentence representation, a word representation, and a feature representation. In representing the sentences, we have only considered the semantic content of sentences; aspects of sentences such as their syntax and pragmatics are not addressed. The propositional information is represented as a number of verb-role-noun triples. The verbs and nouns are the words actually presented in the sentences. The roles are the case roles (Fillmore 1968) that specify the relationships between the verbs and the nouns. In this format, the sentence, "The man stirred the coffee with a spoon," is represented as (stirred agent man) (stirred patient coffee) (stirred instrument spoon)

This representation has several useful properties. First, it allows propositional arguments to be represented independently, since each triple has an individual level of activation. This feature will become important in modelling the argument priming found by McKoon and Ratcliff (1981). Secondly, as we will see, it allows sentences of near arbitrary length to be conveniently represented in a finite structure.

As suggested by the empirical results, words are represented via an underlying, distributed feature-set. The features fall among a number of descriptive dimensions. For instance, the noun features fall among such dimensions as size, animacy, shape, and color. Within each dimension lie several features to represent different values on that dimension. The size dimension, for example, contains three features: small, medium, and large. This partitioning of a dimension allows the certainty of a feature to be represented explicitly. If one were convinced that spoons were small, the small feature could be strongly activated, close to 1, while the large and medium features were kept inactive, close to 0. On the other hand, if one were unsure whether spoons were small or medium, both the small and medium features could be activated to the extent to which the information was believed. If dimensions were represented by the level of activity on a single feature, then the range of activation values a unit could represent would carry the size information and there would be no place to represent the certainty information. Additionally, representing certainty by a level of activation is convenient because strongly

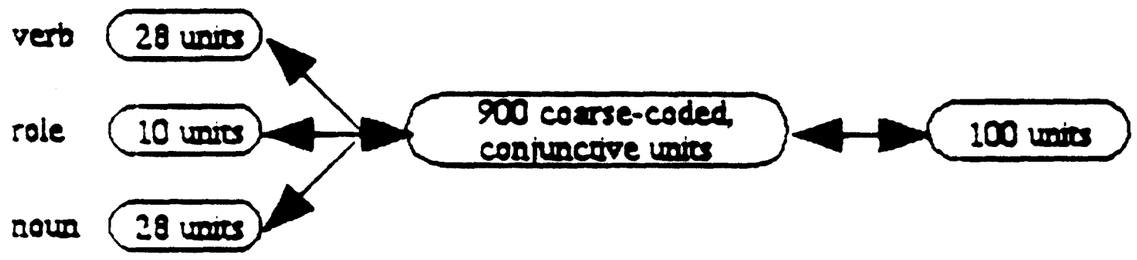
activated units have greater effects on other units than do weakly activated units. Thus greater certainty of information leads directly to stronger interactions.

The prototypical definition of a word consists of one or more active features on each dimension. The activation of each feature can be modified, however, so that this prototypical definition is soon influenced by other words in the input. The prototypical definition can also be incomplete. Container, for example, has no strong, a priori, size specification. This attribute is represented in the model by activating, to a moderate degree, each feature on the size dimension. Thus, no size is certain, but each is possible.

The third representation, the features themselves, are the atomic elements of the model, in that individual features are represented by individual units. To keep the model to a reasonable size, a small subset of all the features needed to represent concepts in the mind was chosen. Further, no attempt was made to insure the features chosen are really used in the mind. The model, therefore, makes no theoretical statement about the choice of features. Instead, the features were chosen to effect two processing criteria. First, since the definition of a concept resides solely in the set of active features, it is critical that this set of features be able to differentiate it from all other concepts. Thus, enough features were used to differentiate the concepts we chose to represent. Secondly, generalization in the model can occur by ignoring the features that differentiate concepts. The fewer the features to be ignored, the greater the generalization. To encourage generalization in semantically meaningful ways, it is important to define similar concepts using similar features, thereby reducing the number of features by which they differ. For example, to encourage greater generalization between man and boy, than between man and table, man should have more features in common with boy than it does with table.

Architecture. The organization of the model consists of a number of pools of units. Input enters the network at the semantic feature layer. When a triple is presented to the network, each word activates units representing its defining features in its appropriate pool.

Architecture



Again, one pool exists for each element in the triple representation. The number of units in each pool reflects the number of features included in the representation of that argument. There are 28 verb feature/units, 10 role feature/units, and 28 noun feature/units. As activation from the verbal code arrives at this layer of the network, top-down processing has not yet begun. This situation creates prototypical, or contextless, encodings in the semantic feature layer. Only at a later point in processing will top-down activation begin to modify these encodings to fit the context of the other elements.

The remaining two layers in the network implement the propositional representation. Only in these layers are the individual elements bound together into triples. This binding function is essential to the performance of the network. Without the binding of elements to their respective triples, the system would be unable to determine which elements belonged to which triples. A coarse-conjunctive encoding (Touretzky & Hinton, 1985) was used to perform this task. Each unit of the encoding represents two features from each semantic pool: verbs, roles, and nouns. The unit will become active if either feature from each pool is active. This representation creates a distributed encoding of each triple consisting of a large number of active conjunctive units.

Eventhough this encoding is able to bind the elements of triples together, it is susceptible to noise. Consider an active unit in the coarse coding layer. It cannot be determined which of the two verb features it responds to is on in the input. Either one, or both, could be on. Similar indeterminacy occurs with the role and noun features. This noise problem, however, is not debilitating to the network's functioning because the noise is distributed evenly across the features. The correct features, the ones actually encoded, are represented in the coarse code much more often than are any distractors.

The final layer of the network helps to bind triples into propositions, and creates and uses the propositional schemas. Units at the coarse code layer are not connected to one another. Instead, each is connected to every unit at the superordinate layer. Each of these units, in

turn, is connected back to the lower layer. Influences from one coarse code unit to another, therefore, occur through the superordinate layer. Units in the superordinate layer are thus able to detect regularities in the activations of the coarse code units. These regularities take the form of patterns of triples and comprise the schematic knowledge in the system. As stated above, the network is able to apply this knowledge to complete triple patterns it has previously encoded. Propositional embellishment, for example, occurs when an incomplete pattern at the coarse code layer is completed by processing through the superordinate layer.

Note that each triple involved in representing a sentence must be active in the model at the same time, but separate networks of units are not required to represent each triple individually. Instead, one network suffices to represent every triple concurrently using the conjunctive code. Eventually, a threshold is reached where additional triples create sufficient noise that the binding of elements to triples begins to break down. Typical sentences, however, are well within the representational limits of the system. A nice feature of the network, then, is that this sort of noise in the system increases gradually as the input becomes overwhelmingly complex, but until this threshold is reached, one network is sufficient for accurate processing.

Learning and processing in the network. Learning in the network is accomplished by modifying the strength of the connections. As with the spread of activation, this process occurs in parallel and only uses information locally available to a unit. Learning proceeds via an error correction process such that when the model produces the wrong output, it attempts to adjust connections to produce the output more accurately.

The goal of the model is to start with various inputs presented at the pull-out net, selects every unit that represents stirring as the verb, instrument as the role, and anything as the noun. The net then determines which concept is most strong the network the elaborated versions of the input. An elaborated version of the input is activated over the semantic feature units. Activation travels through the network and produces a new pattern over those units. On each successive trial the network gradually modifies its connections until it reproduces the input pattern exactly. The details of the learning algorithm can be found in Hinton & McClelland (Unpublished manuscript). When the network is tested, it is presented with an incomplete input pattern, and its job is to produce the appropriate complete pattern.

For example, consider how the network would process the sentence, "John stirred the coffee." At some preprocessing stage, we will assume that the sentence is transformed into the propositional representation used by the network. Namely, (stirred agent John) (stirred patient coffee) The triples are presented to the semantic feature layer of the network one at a time. Each word in the triple activates the appropriate set of features. In turn, these features combine to activate the set of coarse coded units that represent the triple as a whole. The second triple replaces the first, and begins to activate its own features and coarse coded units. Realistically, the network would begin processing the first triple in the propositional layers while the second is loaded into the network in the lower layer. This approach would help implement the principle of immediate processing demonstrated by Carpenter and Just (1983). We have not, to this point, been concerned with this aspect of language processing, however, so we have adopted a simplified procedure to "load" the triples into the network. At this point, the model is only concerned with the results of the processing, rather than its time-course.

Instead of beginning to process triples as soon as they arrive in the propositional layers, the network waits until all of the triples have been presented. In the present case, following the encoding of both triples, units in the coarse coded layer send activation to the superordinate layer. If the network has previously learned about people stirring coffee, it may now bring that knowledge to bear on the input. Assuming that the network has learned about a variety of individuals stirring coffee with spoons, the units in the superordinate layer will attempt to complete the input pattern in the coarse code layer by adding a third triple to the encoding (stirred instrument spoon)

Test measures. There are two ways in which the network can be tested to determine the extent to which it is creating schemas, and using them to infer propositional arguments and concept features. To study the inferential abilities of the network, the relevant measure is the strength with which inferred arguments are activated. This measure could then be transformed into behavioral measures such as reaction times and recall percentages.

In the example above, the concern is the extent to which spoon is inferred to be the instrument of the stirring. The architecture of the network, having each triple concurrently active, makes the task of selecting the noun features involved in representing the inferred instrument difficult. To overcome this problem, the network uses a pull-out net (Touretzky & Hinton, 1985)

capable of viewing individual triples active in the network. The pull-out net works by selecting the coarse coded units involved in representing a triple. Normally, it selects all of the units representing the verb, role, and noun of a specified triple. To determine what instrument the network added to "John stirred the coffee," the pull-out net has to select the units that represent an underspecified triple: the noun is unknown. To handle this problem, the pull-out net, selects every unit that represents stirring as the verb, instrument as the role, and anything as the noun. The net then determines which concept is most strongly activated across the noun features.

First, the coarse-code units selected by the pull-out net send their activations to the noun features they represent. If a particular coarse coded unit represents both round and animate, for example, then its activation is collected by both features. Since the coarse-coding uses random pairs of features, it is likely that one feature of a unit represents the inferred noun while the other feature is noise. Such is the case with the unit representing round and animate. While round is correct for spoon, animate is not: the activation of animate is noise. Fortunately, because each of the large number of features is an equally likely candidate for spurious activation, the number of hits on the correct features is much greater than the number of misses assigned to any particular spurious feature.

The strength to which an entire concept is active is defined as the degree to which each of its features is favored in the pull-out net. Concepts with a number of features in common with the inferred concept will also gain strength. Knife, for instance, differs from spoon on only a single dimension in the current model: its shape. To the extent that spoon is activated on every dimension except the critical one, knife will be activated as well. Spoon wins the strength competition, however, because its shape feature, round, is more active than is knife's shape feature, flat.

This test measure works well for determining which arguments are added to the encoding, and determining which features of concepts are modified. A different measure was needed, however, to model subjects' confidence in answers, such as that found in the Bransford & Franks study. A measurement of the network which seems to capture this idea of confidence is the consistency of the information active in the network: the degree to which the input matches the long-term memory knowledge. This measure is very similar to Hopfield's (1982) energy measure.

If two units are active, and there is a positive connection between them, representing the long-term memory knowledge, then the information is consistent. The positive connection reinforces the receiver's activation. To measure this consistency, we simply multiply the activations by the weight of the connection. As the consistency of the information increases, the product gets larger. By summing the products of every combination of activations and weights in the network, the global consistency of information in the network can be obtained.

A recognition test using a confidence rating is simulated by the network in the following way. The test sentence is presented to the network as a set of triples, just as in testing for inferences. After the network has processed the triples, the consistency is measured. If the input closely matches the long-term memory knowledge, the weights will cause the activations of the units to reinforce each other and produce a high degree of consistency. Otherwise, the activation of units will be set at odds, compete, and thereby lower the degree of consistency.

SIMULATIONS

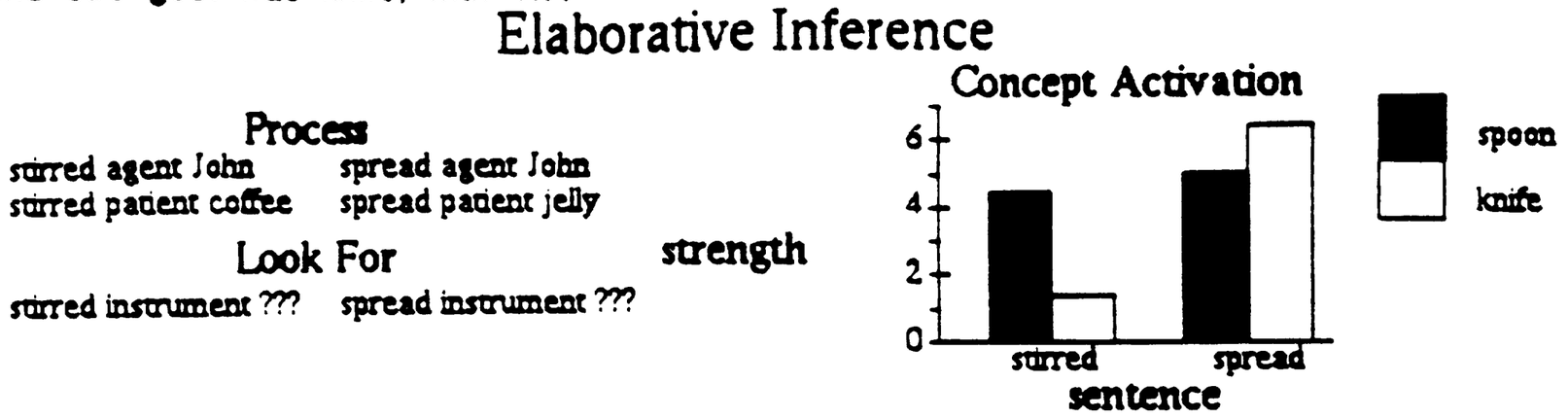
The intent of the simulations was to capture the basic characteristics of the empirical phenomena, rather than to model the particular effects in detail.

Elaborative inference. A basic test of the model's ability to draw elaborative inferences is whether the model can draw the correct inference to suit a context. This effect involves using the context to find the appropriate schema in memory, and then applying the knowledge in the schema to draw the appropriate inference. To implement this test, the model was first taught a number of sentences about people stirring coffee with spoons and spreading jelly with knives. Each person who either stirred coffee or spread jelly did so with the proper instrument. This condition ensured that the correct instrument was strongly related to the action. Specifically, the model learned about three people performing each action. The names of the people, encoded in a six feature/unit subsection of the noun units, were arranged to prevent any correlation between any name and action.

Following the learning of the background, schematic, material, the network was presented

with a test sentence. In one condition, this sentence referred to a new individual stirring coffee, but no instrument was provided: John stirred the coffee.

Did the network infer an instrument, and was it the correct instrument for the context? To address these questions, the pull-out net was used to view the (stirred instrument <noun>) triple. The most strongly activated noun in the triple was spoon, with a strength of 4.7, and the second strongest was knife, with 1.7.

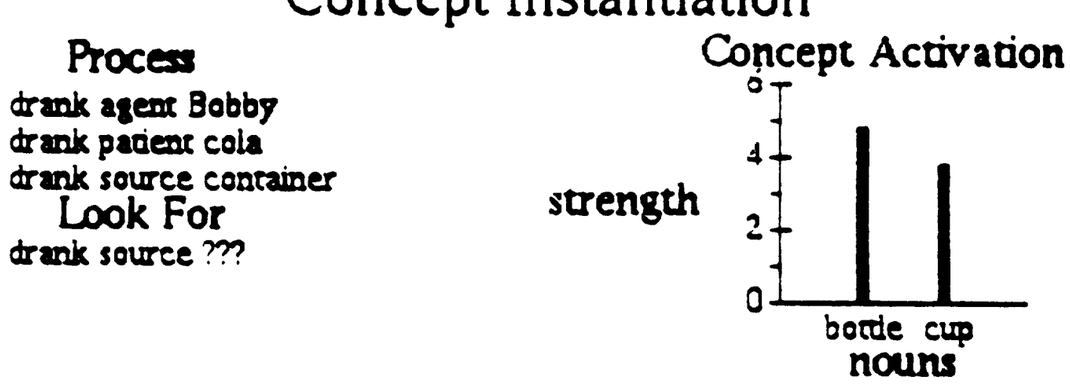


In the other condition, the network was presented with the sentence, "John spread the jelly." Again, no instrument was explicitly mentioned. The most strongly activated noun in this condition was knife, with 6.5, and the second strongest was spoon, with 5.2.

One reason why another utensil was conspicuously active was due to the similarity of its features. As stated above, spoon and knife differ along only the shape dimension. Every other feature of the two nouns both received support as the inferred instrument. However, on the critical dimension, the correct instrument's feature was more active. The fact that the correct instrument was strongest in each condition attests to the network's ability to deliver results in spite of the noise inherent in the architecture. A second reason for the high strength of the alternative was simply that the alternative was a well learned concept consistent with much of the network's long-term memory knowledge. Again, the fact that the network chose the correct instrument demonstrates the model's ability to select the appropriate schema for processing.

The asymmetry in the results can be traced to noise in the coarse-coding. The features of all the nouns in the propositions tend to bleed together somewhat. In the "John spread the jelly" sentence, the "round" feature is hyperactive because another noun in the sentence, John, is also round. Eventhough the amount of noise produced was not enough to interfere with the elaboration effect, we are concerned that the coarse-coding is too noisy a representation.

Concept instantiation. The second set of simulations addressed the network's ability to instantiate general nouns with more specific nouns related to the context of the sentence. The network first learned about a number of children drinking cola from bottles. Once this information had been learned, the network was presented with a new person drinking cola from a container. The encoding of container derived bottom-up from the sentence is specified only for those features common to all containers, such as inanimate, hollow, and man made. The model uses the context to determine how to fill in the context-specific features of the container. The encoding of container was then checked to determine how the concept was in fact instantiated, or refined.

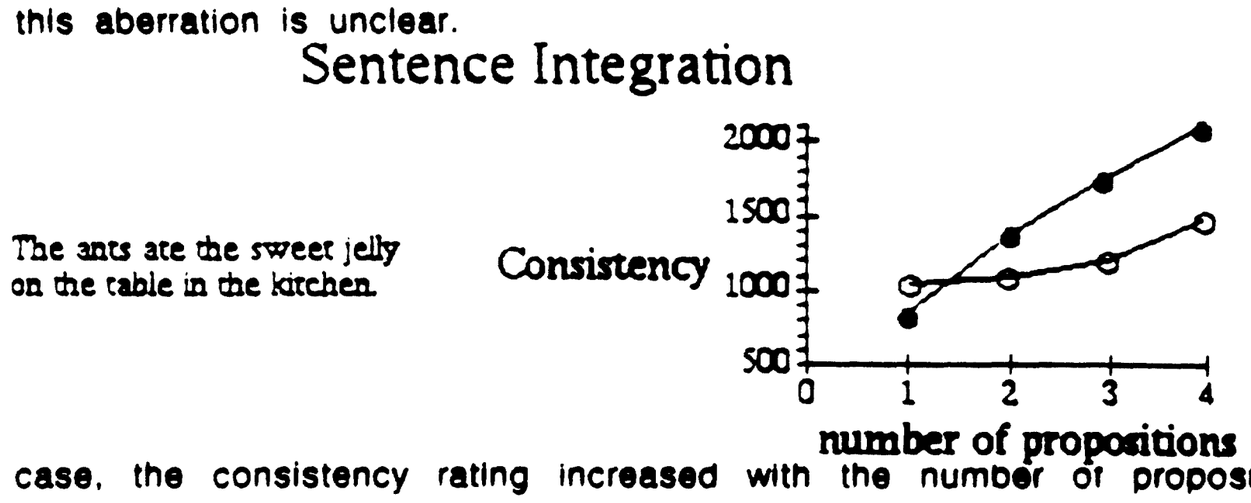


As expected, the network filled in the unspecified features of container to fit the context of the sentence. Specifically the representation of container was modified to look more like the representation of bottle. The strength of bottle in the instrument slot was 4.7, compared to cup, with 2.9.

Proposition integration. The final set of simulations tested the network's ability to create a schematic representation across instances, and then apply that knowledge to a recognition test. In the simulations above, the network was only required to create a schema for a single sentence. In this case, however, a single schema must be created from the combination of a large number of sentences.

As in the Bransford and Franks (1971) study, the network learned a number of sentences that consisted of various combinations of the propositions in a more complex idea. Specifically, the model learned half of the 12 propositional combinations possible in the sentence, "The ants ate the sweet jelly on the table in the kitchen." While there are actually 15 combinations, three combinations do not make sensible sentences. For example, "Ants ate jelly, table in kitchen" cannot be combined sensibly. Of the 12, the model learned half of the single propositions, half of the two proposition combinations, and so on. Following this learning phase, the network was presented with all 12 proposition combinations individually. The consistency of the information in the network was measured after processing each combination to determine the model's confidence in recognition. Next, the model was reset and learned the other half of the propositional combinations. This manipulation provided a counterbalance for the materials so that over the two conditions each combination appeared as both learned and new.

For the sentences the network had studied, the predicted pattern of results was found. Consistency rose with the number of propositions in the test sentence. For the new sentences, similar results were found. The one inconsistency occurred with the single proposition sentences: the new sentences produced a greater consistency rating than the studied sentences. The reason for this aberration is unclear.



In every case, the consistency rating increased with the number of propositions. The positive slope indicates that a similarity judgment is being made based on schematic knowledge created during the study phase. Additionally, aside from the aberration with the single proposition sentences, the previously learned sentences achieved higher ratings. This result demonstrates the model's ability to retain some information concerning the specific examples studied.

CONCLUSIONS

This paper represents a report on work in progress. A great deal remains to be done both in elaborating our understanding of the process of schematic reconstruction through further experimentation, and in applying the model to data. In addition, though the model has several appealing properties, there are limitations that must be overcome before it will be fully extendable. In particular, the model is subject to a greater degree of interference between constituents than seems reasonable; attempts to solve this problem by simply increasing the size of the pool of conjunctive units has not completely solved the problem. For this and other reasons, we are currently at work developing a successor to this model, which uses connectionist learning methods to train both the set of conjunctive units and the pullout net to make more efficient use of each unit.

Despite the incompleteness of the work reported here, some important accomplishments have been made. First, the system is able to add contextually appropriate propositional arguments to the encoding. These inferences are produced by schematic knowledge of the proposition working on the input top-down. Arguments that fit with the pattern of arguments in the input, based on previous experience, are added to the encoding. The system is designed to produce graded effects depending upon the semantic relatedness of the arguments, though this property has yet to be demonstrated clearly in experimental studies.

Second, the system is able to instantiate general concepts to reflect the encoding context. This effect is produced by a distributed feature representation that contains schematic knowledge. This knowledge works on the encoding top-down, modifying concept features to more accurately reflect the schematic knowledge applicable to the encoding context.

Third, the model is able to produce complete propositional representations based on a number of separately presented, incomplete examples. Further, this representation, if used to make similarity judgments to test sentences, gives greater confidence to test sentences with more propositions, as observed with subjects.

The most serious problem was the amount of noise created by the coarse-coding. In the elaborative inference simulation, the degree of noise was particularly high. We expect that the refinements we are developing should alleviate this problem, and should make it possible to store many more patterns in a network of a particular size.

Schematic reconstruction and distributed processing are helpful principles in understanding and implementing language comprehension. Together they help explain a large corpus of psychological effects, three of which are presented here. We have also shown that Parallel Distributed Processing mechanisms provide an effective way to apply these principles, though further research will be required before we will be able to claim that Parallel Distributed Processing mechanisms provide a fully adequate accout of schematic reconstruction processes.

REFERENCES

Anderson, R. C., & Ortony, A. (1975). On putting apples into bottles: A problem of polysemy. *Cognitive Psychology*, 7, 167-180.

Anderson, R. C., Pichert, J. W., Goetz, E. T., Schallert, D. L., Stevens, K. V., & Trollip, S. R. (1976). Instantiation of general terms. *Journal of Verbal Learning and Verbal Behavior*, 15, 667-679.

Bartlett, F. C. (1932). *Remembering: An experimental and social study.* Cambridge: Cambridge University Press.

Bransford, J. D., & Franks, J. J. (1971). The abstraction of linguistic ideas. *Cognitive Psychology*, 2, 331-350.

Carpenter, P. A., & Just, M. A. (1983). What your eyes do while your mind is reading. In K. Rayner (Ed.), *Eye movements in reading: Perceptual and language processes.* New York: Academic Press.

Garnham, A. (1979). Instantiation of verbs. *Quarterly Journal of Experimental Psychology*, 31, 207-214.

Gumenik, W. E. (1979). The advantage of specific terms over general terms as cues for sentence recall: Instantiation or retrieval? *Memory & Cognition*, 7, 240-244.

Hinton, G. E., & McClelland, J. L. (Unpublished). Recirculation.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 2554-2558.

McKoon, G., & Ratcliff, R. (1981). The comprehension processes and memory structures involved in instrumental inferences. *Journal of Verbal Learning and Verbal Behavior*, 20, 671-682.

Reitman, J. S., & Bower, G. H. (1973). Storage and later recognition of exemplars of concepts. *Cognitive Psychology*, 4, 194-206.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, Vols. I & II.* Cambridge, MA: MIT Press.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and sequential thought processes in PDP models. In James L. McClelland, & David E. Rumelhart, *Parallel distributed processing: Explorations in the microstructure of cognition, Vol II.* Cambridge, MA: MIT Press.

Touretzky, D. & Hinton, G. E. (1985). Symbols among the neurons: Details of a connectionist inference architecture. *Proceedings of IJCAI-85, Los Angeles, CA.*

Parallel Distributed Processing and Role Assignment Constraints

James L. McClelland
Carnegie-Mellon University

My work in natural language processing is based on the premise that it is not in general possible to recover the underlying representations of sentences without considering semantic constraints on their possible case structures. It seems clear that people use these constraints to do several things:

To assign constituents to the proper case roles and attach then to the proper other constituents.
To assign the appropriate reading to a word or larger constituent when it occurs in context.
To assign default values to missing constituents.
To instantiate the concepts referenced by the words in a sentence so that they fit the context.

I believe that parallel-distributed processing models (i.e., connectionist models which make use of distributed representations) provide the mechanisms that are needed for these tasks. Argument attachments and role assignments seem to require a consideration of the relative merits of competing possibilities (Marcus, 1980; Bates and MacWhinney, 1987; MacWhinney, 1987), as does lexical disambiguation. Connectionist models provide a very natural substrate for these kinds of competition processes (Cottrell, 1985; Waltz and Pollack, 1985).

The use of distributed representations also seems well suited to capturing many aspects of the way people exploit semantic constraints. For choosing between two distinct alternative interpretations of a constituent, local and distributed representations may be approximately equivalent, but distributed representations are much more natural from capturing contextual shading of the interpretation of a constituent. In a distributed representation the pattern of activation that is most typically activated by a particular word or phrase can be subtly shaded by constraints imposed by context; there is no need to limit the choice of alternative shadings to a pre-specified set of alternatives each represented by a differnt single unit. Similarly, filling in missing arguments is not a matter of choosing a particular concept, but of filling in a pattern that specifies what is known about the filler, without necessarily specifying a particular specific concept.

In previous work, Alan Kawamoto and I (McClelland and Kawamoto, 1986) implemented a parallel-distributed processing (PDP) model that can use semantic constraints to do the four things listed at the beginning of the article, though it was limited to processing only one clause at a time. While it would be possible to use such a mechanism clause-by-clause, semantic constraints are often required to decide which of several clauses a phrase belongs to. For example, in the sentence:

1) John ate the cake that his mother baked at the picnic.

we attach "at the picnic" to the main clause (as the place where the cake was eaten), whereas in

2) John ate the cake that his mother baked in the oven.

we attach "in the oven" to the subordinate clause (as the place where the cake was baked). Clearly these attachments depend on knowing that baking can take place in ovens, not at picnics, and eating can take place at picnics, not in ovens; I would also claim that the relative merits of both attachments must be taken into account to get the attachments right. It seems, then, that a mechanism is needed that can consider the possibility of attaching a phrase to more than one possible clause.

This article sketches out a model that aims to achieve multi-clause capability. The model has not yet been fully implemented, so the paper is quite speculative. However, I think the model promises to take us some distance toward a better understanding of the interaction of syntactic and case-role analysis. In particular, it suggests that with the right connectionist architecture, the four uses of semantic constraints enumerated above become intrinsic characteristics of the language processing machinery.

I would like to thank Geoff Hinton, George Lakoff, Brian MacWhinney and Mark St. John for discussions of the topic of this paper and/or for specific comments on the first draft. Supported by ONR contract N00014-82-C-0374, NR 667-483.

Representing structure and content. To begin, let us consider how to represent the structure of a sentence in a PDP mechanism. To do this, we make use of the notion that a structural description can be repesented as a set of triples. For example the correct role structure of Sentence 2 can be represented with a set of triples such as the following:

(P1 AGENT BOY) (P1 ACTION ATE) (P1 PATIENT CAKE)
(P2 AGENT MOTHER) (P2 ACTION BAKED) (P2 PATIENT CAKE)
(P2 LOCATION OVEN)

An individual triple can be represented in distributed form by dedicating a set of units to each of its parts; thus we can have one set of units for the head of the triple, one for the relation, and one for the tail or slot-filler. Each of the three parts of a triple can then be represented in distributed form as a pattern of activation over the units. The idea of using this kind of three-part distributed representation was introduced by Hinton (1981) to represent the contents of semantic nets; the extension to arbitrary tree structures is due to Touretzky and Hinton (1985) and Touretzky (1986).

For the fillers, or the tail of a triple, the units stand for useful characterizers that serve to distinguish one filler from another. Hinton (1981) used the term "microfeatures" for these units; these features need not correspond in any simple way to verbalizable primitives. Different slot fillers produce different patterns on these units; and the different possible instantiations of a filler are likewise captured by differences in the pattern of activation on the units.

For the relations, the units stand for characteristics of the relation itself. Note that this differs from most other approaches in treating each role or relation as a distributed pattern. This has several virtues. For one thing, it immediately eliminates the problem of specifying a small set of case roles, in the face of the fact that there seem to be a very large number of very subtle differences between roles that are in many ways very similar. Further, the use of distributed representations allows us to capture both the similarities and differences among case roles. The idea has been proposed on independent linguistic grounds, as well.

For the head of each triple, the units stand for characteristics of the whole in which the filler plays a part. Thus the pattern that represents P1 is not some arbitrary pointer as it might be in a Lisp-based representation, but is rather a *Reduced Description* of the constituent that it stands for (Hinton, McClelland, and Rumelhart, 1986; Lakoff, personal communication). In particular, the pattern representing P1 would capture characteristics of the act of eating and of the participants in the act. There would be less detail, of course, than in the separate representations of these constituents where they occur as separate fillers of the tail slot.

Syntactic and case-role representations. Sentences have both an augmented surface structure representation and a case-role representation. In the present model, then, there are two sets of units, one that represents the syntactic structure triples, and one that represents the case-structure triples. I have already described the general form of the case-role triples; the syntactic triples would have a similar form, though they would capture primarily syntactic relations among the constituents. So, for example, the set of syntactic triples of Sentence 2 would be something like:

(S1 SUBJ BOY) (S1 VERB SAW) (S1 DOBJ CAKE)
(CAKE MODIFIER S2)
(S2 SUBJ MOTHER) (S2 VERB BAKED) (S2 DOBJ I = CAKE)
(S1 LOC-PP OVEN)

There are, correspondingly, two main parts to the model, a syntactic processor and a case-frame processor (See Figure 1). In this respect, the model is similar to many conventional parsing schemes (e.g., Marcus, 1980; Kaplan and Bresnan, 1982). The microstructure is quite different, however. One of the key things that a PDP microstructure buys us is the ability to improve the interaction between these two main components.

Syntactic processing. The role of the syntactic processor is to take in words as they are encountered in reading or listening and to produce at its outputs a sequence of patterns, with each pattern capturing one syntactic structure triple.[1] In Figure 1 the syntactic processor is shown in the midst of processing Sentence 2. It has reached the

1. Note that this means that several words can be packed into the same constituent, and that as the words of a constituent (e.g., "the old grey donkey") are encountered the microfeatures of the constituent will be gradually specified. Thus the representation of the constituent can gradually build up at the output of the syntactic processor.

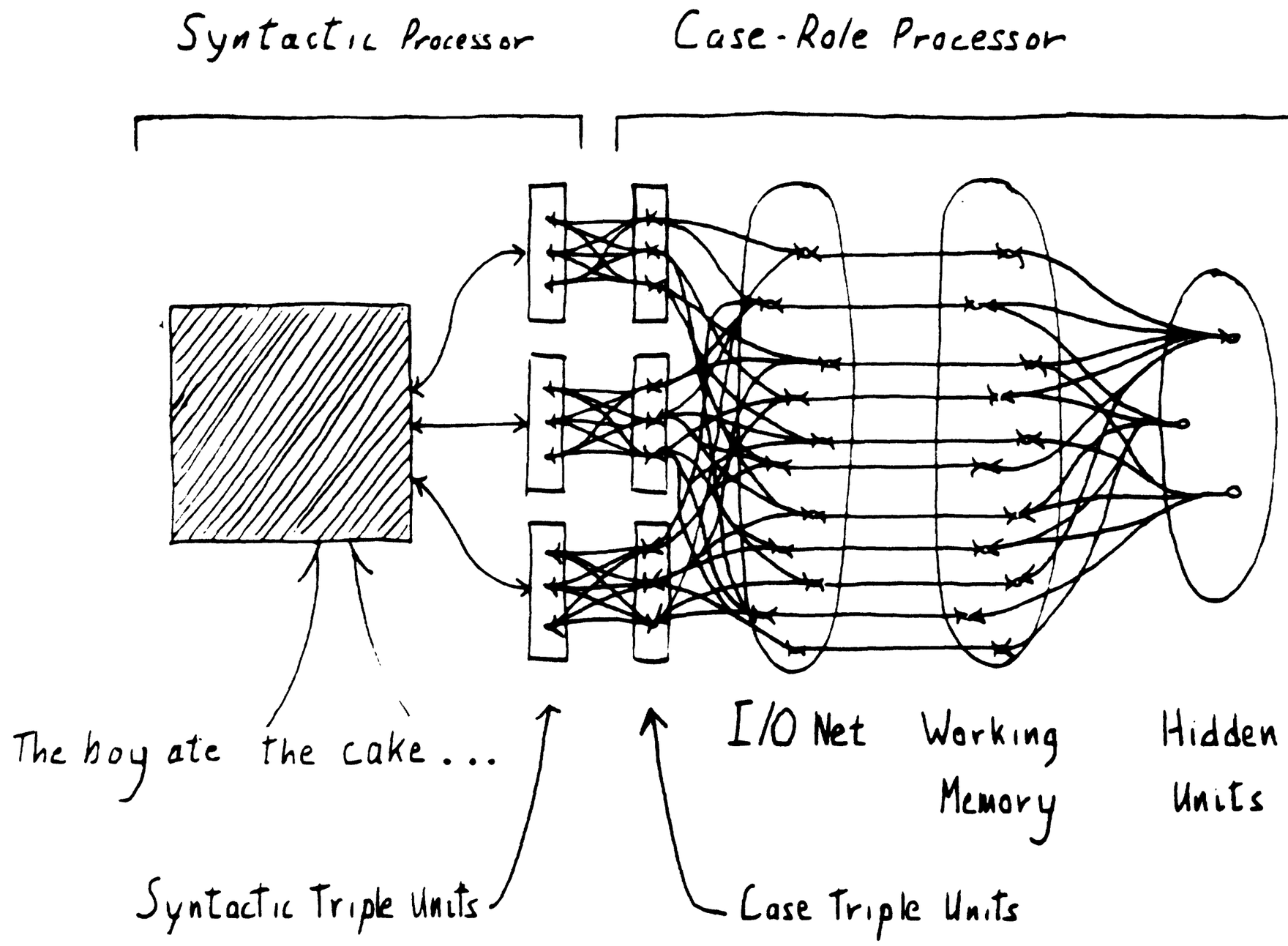


Figure 1. A diagram of the model. See text for explanation.

point where it is processing the words "the cake". The output of at this point should tend to activate the pattern corresponding to (S1 DOBJ CAKE) over a set of units (the *syntactic triple units*) whose role is to display the pattern of activation corresponding to the current syntactic triple. Note that these units also receive feedback from the case-frame processor; the role of this feedback is to fill in unspecified parts of the syntactic triple, as shall be discussed below. The syntactic triple units have connections to units (the *case-frame triple units*) which serve to represent the current case-frame triple.

The connections between these two sets of units are assumed to be learned through prior pairings of syntactic triples and case-frame triples, so that they capture the mutual constraints on case and syntactic role assignments. The inner workings of the syntactic processor have yet to be fully worked out, so for now I leave it as a black box.

The case-frame processor. The role of the case-frame processor is to produce an active representation of the current case-frame constituent, based on the pattern representing the current syntactic constituent on the syntactic triple units and on feedback from a set of units called the *working memory*. The working memory is the structure in which the developing case-frame representation of the sentence is held. As constituents are parsed, they are loaded into the working memory, by way of a network called an I/O net.[2] Within the working memory, individual units correspond to combinations of units in the current case-role representation. Thus, the representation at this level is *conjunctive*, and is therefore capable of maintaining information about which combinations of case-role units were activated together in the same case-role triple when the patterns activated by several triples are superimposed in the working memory (see Hinton et al 1986, for discussion). Of course, early in a parse, the loaded constituents will necessarily be incomplete.

Pattern completion. The working memory provides a persisting representation of the constituents already parsed. This representation persists as a pattern of activation, so that it can both constrain and be constrained by new constituents as they are encountered, through interactions with a final set of units, called the *hidden case-role units*. These units are called "hidden" because their state is not visible to any other part of the system; instead they

2. The I/O net is equivalent to Touretzky and Hinton's (1985) "pull-out net". Its job is to ensure that the characteristics of only a one of the constituents stored in the working memory are interacting with the case-frame triple units. See Touretzky and Hinton (1985) for details.

serve to mediate constraining relations among the units in the working memory. The process works as follows. Connections from working memory units to hidden units allow the pattern of activation over the working memory to produce a pattern over the hidden units. Connections from the hidden units to the working memory units allow these patterns, in turn, to feed activation back to the working memory. This feedback allows the network to complete and clean-up distorted and incomplete patterns (that is, representations of sentences). The connections in the network are acquired through training on on a sample of sentences (see St. John, 1986, for details). The connection strengths derived from this training experience allow it to sustain and complete the representations of familiar sentences; this capability generalizes to novel sentences with similar structure.

What this model can do. The model I have described should be able to do all of the kinds of things listed at the beginning of the paper. Consider, for example, the problem of interpreting the sentence "The boy hit the ball with the bat." This requires both assigning the appropriate reading (baseball bat) and the appropriate role (instrument) to the bat. The syntactic triple for this constituent (S1 with-PP BAT), would tend to activate a pattern over the coresponding to a blend of baseball bat and flying bat as the tail of the triple, and a blend of the possible case-roles consistent with "with" as the the pattern representing the relation portion of the triple. These in turn would tend to activate units representing the various possible filler-role combinations consistent with this syntactic constituent. But since the other constituents of the sentence would already have been stored in the working memory, the completion process would tend to support units standing for the baseball-bat as instrument interpretation more than others. Thus, simultaneous role assignment and context sensitive selection of the appropriate reading of an ambiguous word would be expected to fall out naturally from the operation of the completion process.

Filling in default values for missing arguments and shading or shaping the representations of vaguely described constituents is also a simple by-product of the pattern completion process. Thus, for example, on encountering "The man stirred the coffee", the completion process will tend to fill in the pattern for the completion that includes a spoon as instrument. Note that the pattern so filled in need not specify a particular specific concept; thus for a sentence like "The boy wrote his name", we would expect a pattern representing a writing instrument, but not specifying if it is a pen or a pencil, to be filled in; unless, of course, the network had had specific experience indicating that boys always write their names with one particular instrument or another. A similar process occurs on encountering the container in a sentence like "The container held the cola". In such cases the constraints imposed by other constituents (the cola) would be expected to shape the representation of "container", toward a smallish, hand-holdable, non-porous container; Again, this process would not necessarily specify a specific container, just the properties such a container could be predicted to have.

I have not yet said anything about what the model would do with the attachment problem posed by the sentence "The boy ate the cake that his mother baked in the oven." In this case, we would expect that the syntactic processor would pass along a constituent like (S? in-PP OVEN), and that it would be the job of the case-role processor to determine its correct attachment. Supposing that the experience the network has been exposed to includes mothers (and others) baking cakes (and other things) in ovens, we would expect that the case-role triple (P2 LOC OVEN) (where P2 stands for the reduced description of "mother-baked-cake") would already be partially active as the syntactic constituent became available. Thus the incoming constituent would simply reinforce a pattern of activation that already reflected the correct attachment of oven.

Current status of the model. As I previously stated, the model has not yet been implemented, and so one can treat the previous section as describing the performance of a machine made out of hopeware. Nevertheless I have reason to believe it will work. CMU connectionists now have considerable experience with representations of the kind used in the case-frame processor (Touretzky & Hinton, 1985; Touretzky, 1986; Derthick, 1986). A mechanism quite like the case-frame processor has been implemented by St. John (1986), and it demonstrates several of the uses of semantic constraints that I have been discussing.

Obviously, though, even if the case-frame processor is successful there are many more tasks that lie ahead. One crucial one is the development of a connectionist implementation of the syntactic processor. I believe that we are now on the verge of understanding sequential processes in connectionist networks (see Jordan, 1986), and that this will soon make it possible to describe a complete connectionist mechanism for language processing that captures both the strengths and limitations of human language processing capabilities.

References

Bates, E., & MacWhinney, B. (1987). Competition, variation and language learning: What is not universal in language acquisition. In B. MacWhinney (Ed.), *Mechanisms of language acquisition*. Hillsdale, NJ: Erlbaum.

Cottrell, G. (1985). *A connectionist approach to word sense disambiguation* (TR-154). Rochester, NY: University of Rochester, Department of Computer Science.

Derthick, M. (1986). *A connectionist knowledge representation system*. Thesis proposal, Carnegie-Mellon University, Department of Computer Science, Pittsburgh, PA.

Hinton, G. E. (1981). Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory* (pp. 161-188). Hillsdale, NJ: Erlbaum.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed Representations. In D. E. Rumelhart, J. L. McClelland, & the PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume I.* Cambridge, MA: Bradford Books.

Jordan, M. I. (1986). *Serial order: A parallel distributed processing approach* (ICS Rep. No. 8604). University of California, San Diego, Institute for Cognitive Science.

Kaplan, R., & Bresnan, J. (1982). Lexical functional grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.

Kawamoto, A. H. (1985). *Dynamic processes in the (re)solution of lexical ambiguity*. Unpublished doctoral dissertation, Brown University.

MacWhinney, B. J. (1987). The competition model. In B. MacWhinney (Ed.), *Mechanisms of language acquisition*. Hillsdale, NJ: Erlbaum.

Marcus, M. P. (1980). *A theory of syntactic recognition for natural language*. Cambridge, MA: MIT Press.

McClelland, J. L. (in press.) How we use what we know in reading: An interactive activation approach. In M. Coltheart (Ed.), *Attention and performance XII: The psychology of reading*. London: Erlbaum.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In J. L. McClelland, D. E. Rumelhart, & the PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume II.* Cambridge, MA: Bradford Books.

St. John, M. F. (1986). *Reconstructive memory for sentences*. Working paper, Department of Psychology, Carnegie-Mellon University, Pittsburgh, PA.

Touretzky, D. S. (1986). BoltzCONS: Reconciling connectionism with the recursive nature of stacks and trees. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*. Amherst, MA, 522-530.

Touretzky, D., & Hinton, G. E. (1985). Symbols among the neurons: Details of a connectionist inference architecture. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*.

Waltz, D. L., & Pollack, J. B. (1985). Massively parallel parsing. *Cognitive Science, 9*, 51-74.

THE BASIS OF LAWFUL BEHAVIOR: RULES OR CONNECTIONS?

What is the basis of lawful behavior? What knowledge underlies it, and how is it acquired? My colleagues and I have been working toward a new kind of answer to these questions. We have discovered that lawful behavior can emerge from the performance of a network of simple computing elements. We have also discovered that these networks can learn to behave lawfully through experience.

To illustrate, let us consider a simple kind of lawful behavior: the productive use of the past tense of English. Even reasonably young children can form the past tenses of familiar words in English. More than this, they can form the past tenses of made-up forms that they have never heard. Jean Berko demonstrated this in experiments on young children in 1958. Even more strikingly, young children often regularize irregular words; they say things like "taked" and "goed". Berko took this evidence of the productive use of the past tense as evidence that the child had acquired the rule. To quote her 1958 paper:

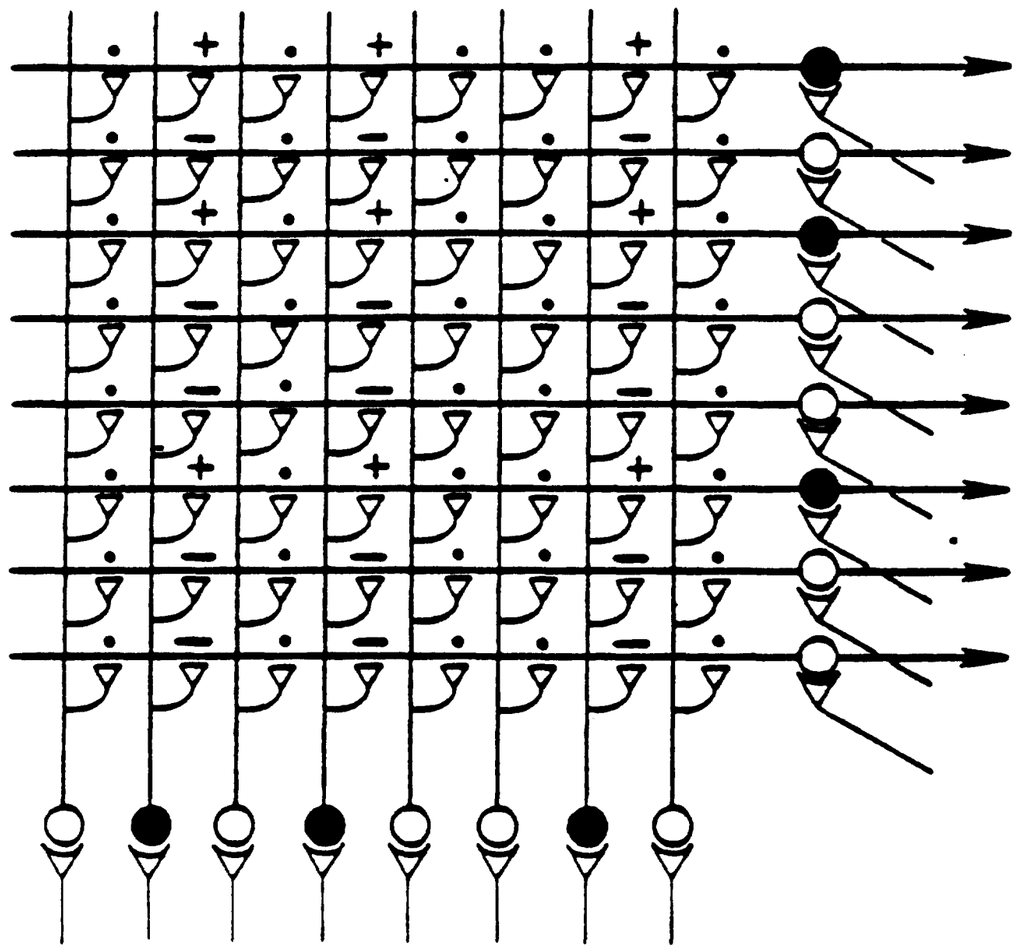
> If a child knows that the plural of witch is witches, he may simply have memorized the plural form. If, however, he tells us that the plural of "gutch" is "gutches", we have evidence that he actually knows, albeit unconsciously, one of those rules which the descriptive linguist, too, would set forth in his grammar.

Berko's argument sounds reasonable, but on close scrutiny a question arises. Exactly what is the form of the unconscious knowledge of the rule? Is it written down in the mind in some sort of explicit form, simply inaccessible to overt report? Do the processing mechanisms actually consult these rules, and do the learning mechanisms actually formulate, evaluate, and/or modify members of the rule set?

My colleagues and I have begun to develop an alternative to this type of account. In our view, the implicit knowledge is stored in

From: The Computers and Philosophy Newsletter, Issue 2:1, Jan 1987, Center for Design of Educational Computing, Carnegie Mellon University.

Figure 1



A very small connectionist network, consisting of two groups of units like those used in our simulations of past-tense learning. The input units are arranged in a row along the bottom of the figure; the output units are in a row down the right-hand edge; the connections among the units are indicated in the square array. The "+", "-", and "." symbols above the connections indicate excitatory, inhibitory, and null connections respectively. These particular connection strengths would allow the indicated input pattern (dark circles along the bottom) to produce the indicated output pattern (dark circles along the right edge). From D E Rumelhart and J L McClelland, "On learning the past tenses of English verbs", in J L McClelland, D E Rumelhart, and the PDP research group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol 2: Psychological and Biological Models.* (Cambridge MA: MIT Press/Bradford Books, 1986). Reprinted with permission.

connections among simple processing units organized into networks. While the behavior of the network may be describable (at least approximately) as conforming to a system of rules, the network models have properties that differ from explicit formal rule systems. These properties allow them to capture several important characteristics of the language acquisition process, as we see it occurring in the human language learner.

To give you the flavor of our approach, I will describe a computer simulation model David Rumelhart and I have developed that learns to produce past tenses of English verbs from exemplars. The model greatly simplifies the past tense learning task, compared to the task as the child confronts it, and isolates it from the rest of language acquisition. These simplifications allow us to focus on the basic point, which is to illustrate how lawful behavior can be acquired by a network.

In our version of the task, the model is presented with training pairs, consisting of the present tense form of a word, paired with the corresponding past tense form. Thus it might be shown go-went, like-liked, etc. Its task is to learn to produce the appropriate past tense form, given the root form as its input.

The model consists, primarily, of two sets of simple computing elements. (See Figure 1.) Each element is a very simple device that takes on an activation of 0 or 1, based on the weighted sum of inputs from other units. One of these networks is used to represent the root form of the word, and the other is used to represent the past tense form.

Processing works like this. When a root form is presented, it produces a pattern of activation over the root form units via an encoding network that translates the sequence of phonemes into a pattern of activation. Each of the root form units represents a phonological property, and if a unit is turned on, we can think of this as indicating that the property it stands for is present in the root form of the word being processed. There is a large number of units, and each word turns on a large subset of them. The representations of different words overlap with each other in this representation, in that they share many properties -- but each word has its own unique set of properties that represents it.

Now in most models, representations can be seen as patterns, but in these models, they are patterns of a particular kind -- they are

active patterns that can activate other units through connections. Each of the units in the root network has a connection to every unit in the past tense network, and whenever a unit is on it sends signals to all of the units it is connected to. These signals are weighted by the connections, which may be positive or negative. If positive, they tend to turn the receiving unit on; if negative, they tend to turn it off. The receiving units add up the signals they receive, and if the net input is strongly excitatory they come on with high probability; if it is strongly inhibitory they stay off with high probability. Intermediate values produce intermediate probabilities of the unit coming on.

Now it turns out that this kind of network can be trained to find values of the connections from one set of units to another so that an arbitrary pattern on the input units will produce a particular output pattern on the other set of units. The training procedure is very simple. We just present the input pattern and allow it to produce an output pattern based on the current values of the connection strengths. Then for each output unit, we compare the obtained pattern with the desired one. When a unit is not active that should have been, we increase the strength of the connections coming into it from each active input unit. This means that next time the same input will be more likely to turn this unit on. When a unit is active that should not have been, we decrease the strength of the connections coming into it from each active input unit. This means that next time the same input pattern will be less likely to turn this unit on. If we repeat this procedure repeatedly with the same pattern pair, we can guarantee whatever level of accuracy we wish with this procedure. In fact we can train a network to respond correctly to all the members of a large ensemble of patterns in this way (as long as certain technical conditions are met).

Now, think with me about the following experiment. Suppose we train the model with a set of patterns that all exemplified the regular past tense pattern of English. That is, we present successive pairs like "like-liked", "hate-hated", "love-loved", etc. For each, we present the root form, we test to see what the network generates, and we adjust the connections wherever there are discrepancies between the obtained output and the correct past tense form. The network will develop strong connections from input features to the corresponding output feature. Initial "l" in the input will activate initial "l" in the output, etc. In addition, it will learn to add the correct, "regular"

Table 1

Correspondences between the Simulation Model and Acquisition Data

Anticipated:

1. Model exhibits over-regularization responses ("go" -> "goed").
2. The model exhibits variability in its responses during transitional phases of acquisition ("go" -> "goed" coexists with other responses).
3. The transition to the adult state is very gradual (regularization errors persist well into grade school, becoming less and less frequent).
4. The "penetration" of the "past-tense rule" is less than perfect; children are better at using it with familiar words than with novel forms, even as late as third grade.

Unanticipated:

5. A special type of transition error, in which irregular past tense forms are combined with the addition of the "-ed" ending, enter processing late in the transitional phase when regularizations are only occurring about 10% of the time (examples are "wented" and "ated").
6. Among irregular forms, those involving no change in forming the past tense (e.g., "hit", "bid") are easiest to learn.
7. Correspondingly, monosyllabic verbs ending in "t" or "d" that should have "-ed" added, tend to be used in past-tense contexts with no change (this includes made up verbs like "mott" as well as real ones like "pet" as in "he petted the dog").
8. Irregular verbs involving vowel changes only are regularized more than irregular verbs involving vowel change and a final consonant change (e.g., verbs like "sing" are regularized more than verbs like "seek").

past-tense ending.

While the pattern of connections is built up from experience with particular exemplars, the model comes to be able to act in accordance with the past tense rule. Not only can it correctly form past tenses of words in the training set, but it can also do very well on the past tenses of words it has not seen before.

OK, you say, so that's a mechanism that learns to act in accordance with the past tense rule, but so what? Why should I believe the mind really works this way, instead of in terms of some real rule induction process? And anyway, even if I accept that it really does work this why, why shouldn't I just ignore this and treat the model as a statement about the implementation details? After all, the mechanism behaves just as if it did have the rule, doesn't it? What difference does it make?

It makes a lot of difference. For mechanisms like this have a lot of properties that correspond to what we see in the language learning of young children. First of all, the mechanism is not thrown by noise -- in this case exceptions in its inputs. It can learn, gradually, to find a set of connections that captures both the regular pattern and the exceptions in the same set of weights. Early on in learning, if it receives a small number of exceptions mixed in with a large number of regular verbs, it learns the regular pattern and overregularizes the irregular forms. As I mentioned before, we see this phenomenon of overregularization in the past tense usage of young children.

Rumelhart and I have run several simulation experiments using training lists consisting of a mixture of regular and irregular verbs. These simulations exhibit a number of features that are characteristic of the acquisition of the past tense of English. One might think that something as simple as the past tense would not be a rich field of empirical evidence, but in fact it is. In Table 1 I have enumerated several aspects of the models' behavior that are actually observed in the speech of children learning English as their first language.

I am very enthusiastic about this model, but I don't want to give the impression that I think it is perfect. It does have flaws, but these are due I think, to the simplifications that we incorporated to illustrate the basic point that lawful behavior could emerge from a network of simple processing units. Rather than dwell on how we intend to improve the model, I will return briefly to the basic issue.