PARALLEL RETRIEVAL AND APPLICATION OF CONCEPTUAL KNOWLEDGE

Charles P. Dolan
Michael G. Dyer

February 1988
CSD-880010

Parallel Retrieval and Application of Conceptual Knowledge[1]

Charles P. Dolan
Hughes AI Center
UCLA AI Laboratory
cpd@cs.ucla.edu

Michael G. Dyer[2]
UCLA AI Laboratory
dyer@cs.ucla.edu

1. The need for conceptual information in connectionist networks

Humans have the ability to recognize and reason about a wide variety of social behaviors in their fellows: deception, professional relationships, goal seeking, argumentation, etc. Much of this ability is centered around our ability to manipulate conceptual knowledge. A recent compendium of psychological results relating to human use of these structures is (Galambos, Abelson, and Black 1986). There have also been a number of computer programs[3] which have demonstrated the capability to use conceptual knowledge as modeled by data structures in symbolic languages. Three fundamental problems are knowledge representation, knowledge access (i.e. which knowledge structures to activate) and knowledge application to specific instances (i.e. which symbols or roles in the knowledge structures are bound to which objects in the current situation). Although results have been enlightening as to processes underlying the use of conceptual knowledge, they have all been based on serial examination of symbolic data structures. This serial approach is unacceptable as a complete model of human cognitive performance.

1.1. Computer model of conceptual information storage and retrieval

Early models of conceptual knowledge concentrated on the ability of knowledge to "control inference and direct search" (Schank 1987). Most research prototypes constructed to date do not have enough knowledge in memory to cause knowledge access to become a serious performance issue. Nevertheless, models of how knowledge is indexed, such as dynamic memory theory (Schank 1982), have been put forth to try and shore up this deficiency in these models. The original dynamic memory theory was posited to account for a large number of aspects of how memory is accessed and updated, specifically: reminding, memory confusion, forgetting, and generalization. The actual embodiments of the theory in programs, however, have stressed only two aspects: categorization(Lebowitz 1983) and elaboration (Kolodner 1983). The main strengths of the theory have been in explaining cross-contextual reminding using thematic knowledge structures (Seifert *et al* 1986) and in modeling confusions in episodic memory (Bower *et al* 1979). However the theory has not

---

[1] Prepared for the AAAI Symposium, "How can Slow Components Think so Fast?", March 22-23, Stanford University.

[2]The second author is supported in part by the JTF program of the DoD and by a grant from the Hughes Artificial Intelligence Center.

[3] The programs using conceptual knowledge are too numerous to summarize in this abstract but a number of references to modeling work which relates to the experiments in the aforementioned collection can be found in (Schank 1987).

been able to account for effects such as priming, retroactive inhibition, associative recall and more general types of learning.

1.2. Computer models of spreading activation and knowledge application

Previous work on parallel application of knowledge has examined one of two problems: semantic retrieval or variable binding. Semantic retrieval has mostly been attacked through the use of spreading activation models (after Quillian 1968) which search a network of symbols in parallel. Two types of spreading activation have been posited, either passing discrete symbols along the network such as Charniak's (1986) or Fahlman's (1977) marker passing techniques, or the spread of continuous activation levels such as in Anderson's ACT* (1983) model. Both these techniques can account for strong, proven, semantic priming effects. Unfortunately, both techniques also require a serial component to either explicitly evaluate paths found or implicitly apply knowledge structures activated by the spread of activations. The reason for this serial step is that neither method can accomplish variable binding; therefore, constraints among symbolic roles (such as equality constraints) cannot be enforced during the spread of activation and so paths must be evaluated serially. This step also requires a change in representation from a propositional network to a predicate calculus (Charniak 1986) or production system (Anderson 1986) format.

1.3. Neuromorphic work on variable binding

Two types of concept retrieval have been demonstrated in distributed connectionist networks[1]: (1) structured object retrieval and (2) micro-feature based associative retrieval. Touretzky and Hinton's (1985) parallel production system has demonstrated various types of parallel constraint satisfaction to perform role binding. These techniques use direct representations of symbols, e.g. each symbol is represented using a feature vector or micro-features. Architectures built using these distributed representations are capable of performing pattern matching where the patterns contain variables but they do not account for purely semantic retrieval. That is, a structure which does not match the variable bindings in long term memory will not be retrieved. In micro-feature based associative retrieval, the pattern of activation which is closest in absolute terms (e.g. hamming distance) to the input pattern is replicated or in some models a combination of two patterns which closely matches the input will be retrieved (Rumelhart and McClelland 1986).

For some tasks, such as learning the past tense of verbs, micro-feature based associative retrieval is exactly what one wants, and for other tasks, such as modeling human puzzle solving, purely structured object retrieval is what one wants. However, when a system is trying to process conceptual knowledge, we would like it to be able to do both. What are the desired characteristics of a realistic model for retrieval and application of conceptual knowledge? First, we must have a model which accounts for psychological data on cued retrieval, but does not require a change in representations (such as moving to predicate calculus) to apply the knowledge. Second, we require a model which is able to create and apply role binding constraints in addition to associative retrieval for accessing conceptual knowledge. For example, Seifert and her colleagues (1986) have shown that people are susceptible to thematic priming where there is little semantic similarity between the cue and the test. In previous work (Dolan and Dyer 1987) we had proposed that a sub-symbolic model of conceptual knowledge use would not require a *general variable* binding mechanism such as that in (Touretzky and Hinton 1985), but only perform *role* binding.

---

[1]Concept retrieval is also performed in local connectionist networks (Shastri and Feldman 1984) but that work is aimed primarily at evidential reasoning (Shastri and Feldman 1985).

Role binding is the process of propagating change in one role or slot of a frame to all other slots which point to the same object. However, we have found that in order to model results such as those of Seifert and her colleagues we have had to admit a more general variable binding mechanism. An example of the need for variable binding in conceptual knowledge is the concept of "flattery", getting someone to do something for you by complimenting them. The need for variable bindings in this concept arise from the need to represent the fact that the mental state which the flatterer seeks to evoke in the listener will service one of his goals. This representation requires that one structure point to another, but when the structure is stored in long term memory there is no way of knowing initially what type of object the structure will need to point to because many different statements could constitute a compliment.

2. Changing the assumptions and criteria for symbolic distributed processing

A step in the right direction would be a distributed model which is able to handle the same symbolic schemata used in other models of conceptual knowledge. Symbolic schemata have been used extensively in story understanding and planning under various names: scripts (Cullingford 1981), sketchy scripts (DeJong 1979), MOPs (Lebowitz 1983) and plan boxes (Wilensky 1983). Schema recognition is a fundamental operation in such systems. Role binding has also played a prominent role in such systems, especially in the BORIS system (Dyer 1983). Some connectionist systems (Golden 1986, and Rumelhart *et al* 1986) have attempted to incorporate ideas of schema recognition but these systems have suffered from an impoverished representation of schemata compared to those used by symbolic systems. Figure 1 shows an example schema with some unfilled roles: 'waiter', 'owner', 'food', and 'payment'. This schema is simplified and adapted from the restaurant script in (Schank and Abelson 1977). The notation used in Figure 1 stresses that a schema-based representation can be implemented as a set of relations, one relation for each slot in the schema. Likewise, the constraints among the slots of the schema, which constitute the structural description of the schema, can be represented as sets of relations.

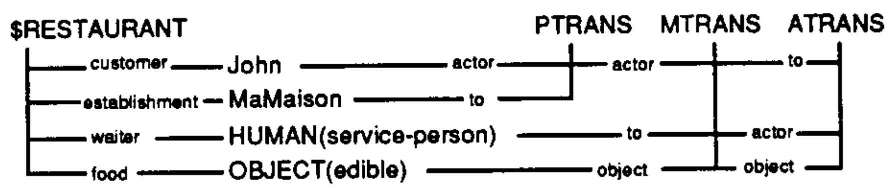

Figure 1: Sample conceptual structures, restaurant

All previous models of conceptual knowledge have operated under the physical symbol system hypothesis (PSSH) (Newell 1980). In elaborating this hypothesis, Newell laid out a candidate list of a "...baker's dozen...[of]...constraints on the nature of mind-like systems...". Systems which meet the strict definition of a physical symbol system (formally equivalent to a Turing machine) automatically meet two of these constraints: (1) universality and (2) symbolic behavior. Furthermore, at the time there was hope that "an accepted generative class of systems that are universal-symbol and also rational" would grow from concepts such as goal hierarchies and heuristic search, thus satisfying another of Newell's constraints on mind-like systems. At this stage in the development of neuromorphic systems, we feel that it is time to pursue systems which not only satisfy the above three constraints but also three more: (1) behave robustly in the face of error, (2) be realizable within the brain and (3) operate in real time. To this end we have concentrated on giving connectionist networks the ability to use conceptual knowledge. As an example, we give a brief description of such a system.

3. Representation and architecture

One way to think of symbols in distributed connectionist models is as "bit strings". These "bit strings" are not memory addresses, as are symbols in traditional implementations, but are feature vectors. An example of this is found in (McClelland and Kawamoto 1986) where each symbol is classified along a number of dimensions. A fixed number of bits is allocated to each dimension, one for each possible classification, and the bit string is formed from that set of features. As an example from (McClelland and Kawamoto 1986),

```
DIMENSIONS    FEATURES
  GENDER   male,female
   SIZE    medium,large
```

are the features, and the symbol mappings are,

```
SYMBOLS          VECTORS
               GENDER SIZE
John     -->    ( 10    01 )
Mary     -->    ( 01    10 )
```

To keep the descriptions simple, we will reuse the bit patterns for the symbols above in our description of frames and slot names. For example, the pattern for a frame and two of its slots might be:

```
HIT     -->    (1010)
actor   -->    (1010)
object  -->    (0101)
```

By constructing symbols this way, we ensure that symbols with similar meanings will have similar representations. Relations can then be formed from these symbols in a very straightforward manner, as in (Hinton 1981) by representing a relation as concatenations of bit vectors. For examples, we can represent "John hit Mary" with two relations `(1010 1010 1001)` for `(HIT actor John)` and `(1010 0101 0110)` for `(HIT object Mary)`. In Hinton's terminology the three symbols are called `(ROLE1 REL ROLE2)`, but in more traditional terminology they would be `(FRAME slot Filler)`.

Sets of relations of these units can be represented on a set of units using conjunctive coding (Hinton *et al* 1986). An example of such a representation is given in Figure 2. Here we allocate a cube of units where each dimension of the cube is the length of one symbol. Each unit in the cube represents a three-way conjunction of the features of the ROLE1, REL, and ROLE2 positions of a relation. In this way, multiple relations can be stored on the same set of units. In the figure, the active units for the triple (HIT actor John) are shown in black.

This approach is similar to the design of the working memory for the production system in (Touretzky and Hinton 1985), except in that design, each element of the working memory was assigned a random subset of each of the possible symbols for the three positions in a relation. In our design, each unit in the working memory plays a very specific role in the semantic meaning of the relations stored in working memory. The advantage of using this approach is that it is easier to design networks which make use of many modules, because the modules have a regular structure. Also, the results of a learning algorithm (that adapts to the pattern representing a concept on a conjunctive cube) will be easier to analyze when the structures are regular. The primary disadvantage of this approach is that ambiguities in the memory develop more quickly than with the random subset approach.

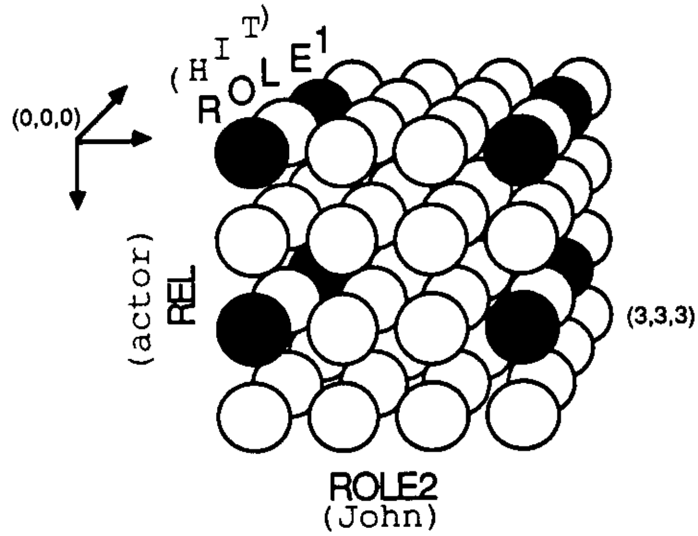

Figure 2: Relations are represented on a conjunctive cube

From the discussion above, we can see that distributed representations impose a different set of constraints than the PSSH. First, it is difficult to make arbitrarily fine distinctions among symbols, because similar symbols have similar representations. Second, in a distributed representation, types and tokens are almost impossible to distinguish, a fact which has been cited as a weak point by Norman (1986) but as a strong point by Rumelhart and McClelland (1986). Third, it is easy to decide if two symbols are similar. Finally, connectionist memories are limited in the number of patterns they can store (Dethrick and Plaut 1986).

Even though some of the assumptions of the PSSH change, others remain. For example, we still have symbols which are manipulated by the system, and we can "open up" a symbol's definition by bringing into working memory all the relations which have that symbol in their ROLE1 position. In addition we still have the assumption that a symbol's definition may only reference other symbols and therefore only reference that symbol's definition indirectly. The major change is that more processing can be done without "opening up" a symbol's definition because the representation of a symbol has some semantic content (instead of being merely an address bit vector).

We have constructed and simulated a model which partially satisfies all of the requirements we have outlined in Section 1.3 for a system which retrieves and applies conceptual schemata. It uses the encoding of symbol structures described above. This means that the representation is also compatible with our previous model of binding new information to instantiated knowledge structures (Dolan and Dyer 1987). We have performed experiments with this architecture in selecting the most appropriate symbolic structures give a partial description of a situation. Although activation flows through the network in parallel, the model effectively uses associative retrieval on semantic features first to narrow the possibilities (as in the parallel intersection search used in marker passing) and then uses role binding constraints to make a final selection when semantic features are not sufficient. If no correct binding can be found then one of the semantically related structures is retrieved. In summary, the model exhibits the behavior of both spreading activation and parallel constraint satisfaction, whichever is appropriate, without any change in representational format as in (Charniak 1986).

The processing architecture is shown below in Figure 3. The connections from short term memory (STM) to other processing units that support story comprehension and planning have been discussed elsewhere (Dolan and Dyer 1987). The long term memory (LTM) consists of winner-take-all clusters modeled after Smolensky's (1986) "knowledge atoms". The conceptual structures in LTM are encoded on the weights between LTM and the retrieval buffer. There is no direct connection between STM and the retrieval buffer. The only connection is via the mapping switch. If there were a direct connection the network would only retrieve knowledge based on semantic features.

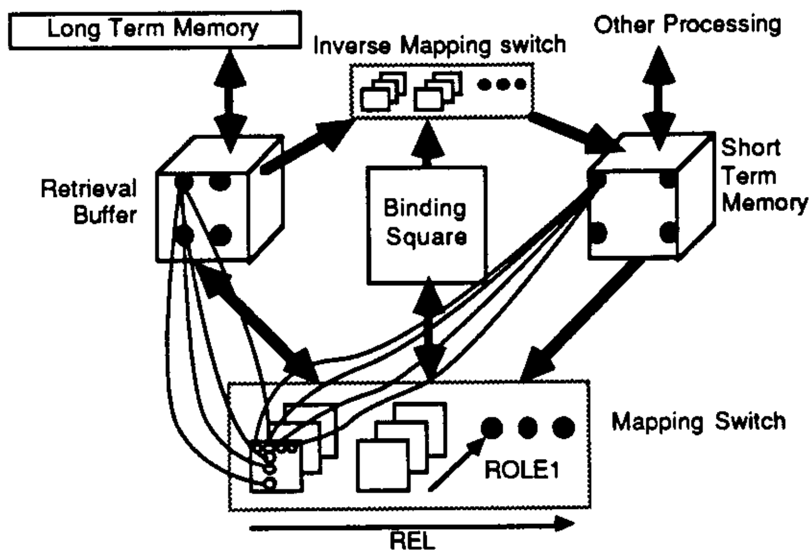

Figure 3: Architecture for conceptual knowledge retrieval

To understand the operation of this architecture, one must notice that in the same way that a relation among three symbols can be represented on a cube of units, a relation among two symbols can be represented with a square, and that a variable binding is essentially a relation among two symbols, one in short term memory and one in the retrieval buffer. Therefore the mapping switch is best conceptualized as a set of squares, where each square represents a partial binding. In the binding switch there is a square for each conjunction of a single bit in the ROLE1 and REL symbols. The activity of a unit in the mapping switch signifies that a particular conjunction of a ROLE1 and REL (frame name and slot name) maps one bit of the ROLE2 representation in the STM onto another bit in the retrieval buffer for that same ROLE1 and REL. Figure 4 shows[1] the details of some of the connections among STM, the mapping switch, the binding square and the retrieval buffer. Only single connections are shown between units. The bi-directional arrows in Figure 3 indicate symmetric connections among the units shown in Figure 4.

[1]Figure 3 shows the architecture for a representation with three bits for each of ROLE1, REL and ROLE2. The largest network we have simulated to date has used eight bits for each of these. Details on the performance of these networks can be found in (Dolan, forthcoming).

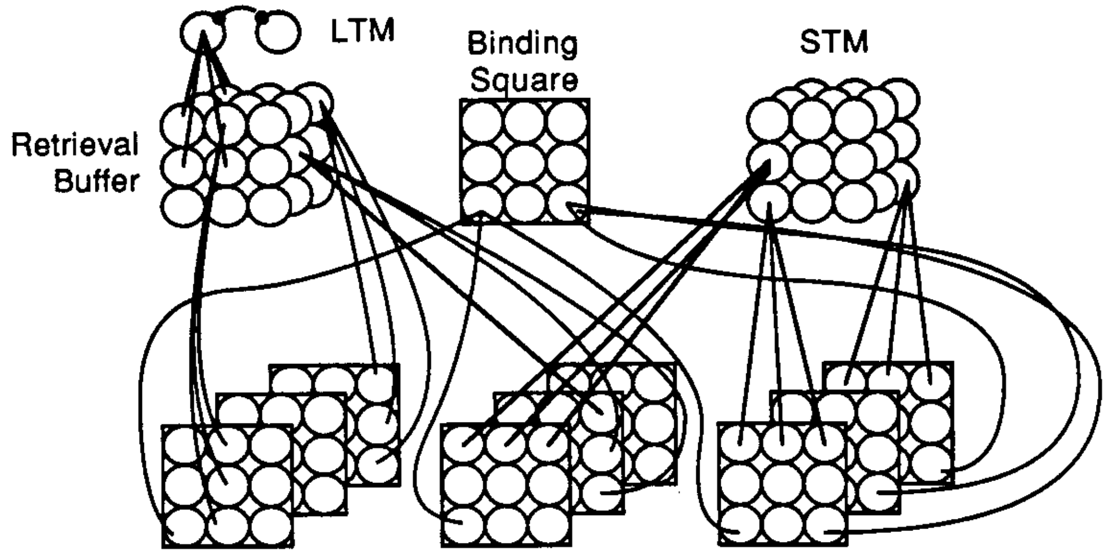

Figure 4: Details of the connection topology

Within the mapping switch incompatible mappings are connected with mutual inhibition along the rows and columns of the partial bindings. Each of these mapping units, however, only determines if it is locally consistent; the network as described so far can still contain many global inconsistencies. Maintaining local consistency ensures that each frame and slot name map a bit in the STM to exactly one other bit in the retrieval buffer and vice versa. Maintaining global consistency ensures that two entire schemata are correctly bound at the symbol level. The binding square attempts to resolve these global inconsistencies by computing the distributed representation of the bindings between the symbols in STM and the variables LTM and using this to resolve inconsistencies in the mapping switch. The activity of a unit in the binding square signifies that a particular mapping of one bit to another is globally consistent. The binding square can be thought of as counting up the votes in the mapping switch.

The function of the binding square together with the mapping switch is similar to the "bind units" in Touretzky and Hinton's (1985) production system. The major difference is that this architecture does not require a fixed concept size. The reason it can handle some variation in concept size is that inconsistencies are constrained via direct inhibitory connections between semantically inconsistent units and not via global inhibition which *a priori* constrains the number of active units in a module. As in the mapping switch, inconsistent bindings[1] in the binding square are connected with mutual inhibition along the rows and columns, therefore, if there are 'n' bits in the ROLE2 representation, only 'n' bits can be fully active in the binding square. This places some limitations on the capacity of the network but not a hard limit. A structure can still be retrieved, even if only part of its mapping is present. The network degrades gradually as the bindings become more complex.

In the first few cycles of retrieval, the LTM units for structures which have substantially the same conjunction of frames and slots in the STM get much more activation than others. At

[1]The inconsistencies are structural not semantic.

this point, most of the computation has been due to the action of the STM on the retrieval buffer via the mapping switch. Neither the inhibitory connections in the binding switch nor the binding square have had much effect.

After the most likely candidate structures are partially instantiated in the retrieval buffer, the binding square makes a few decisions on particular bits being mapped to one another globally. This information feeds back into the mapping switch and allows some the candidate structures to be eliminated. As this process continues, the binding square settles on global mappings and alternate candidates are eliminated until a complete mapping is determined and a single conceptual structure is instantiated in the retrieval buffer.

Once the bindings have been determined, the inverse mapping is accomplished with another set of switching units (i.e. the inverse mapping switch) which is isomorphic to the mapping switch but which simply ANDs together the output of the binding square and the retrieval buffer. This set of units then excites the units for the re-mapped structure in STM.

Of course the operation of the network is not as smooth as the preceding discussion implies (although this is in general the manner in which it settles into a stable state). The LTM, the mapping switch and the binding square are each connected like a Hopfield optimization network (Hopfield and Tank 1985). However, the dynamics and hence the energy surfaces are greatly complicated by interactions with the other component networks. We can get some insight into dynamics if we look at the types of local minima to which it is subject. The most common type of local minima is where a row or column of the binding square settles on intermediate values for two units instead of a single highly active unit. Another type of local minima is when more than one unit representing a conceptual structure remains active in the LTM. This problem is aggravated when the units in the LTM are replaced by cliques of units as were used in Touretzky and Hinton's (1985) distributed production system. The most common type of local minima is found when there is a semantically similar but structurally dissimilar concept in the network. If the network is forced to settle too quickly, then it may find one of these. The problem is that, at a distance in the hyper-space of unit activations, the gradient towards a structure with correct bindings is not very different from semantically similar ones with incorrect bindings. If the network moves towards the two minima too quickly, a small amount of noise will cause it to select the wrong concept. More simulation work is required to determine how these local minima can be avoided and how their number and character change with the increase in complexity of conceptual structures.

4. Conclusions

If the ability to retrieve and apply conceptual knowledge is a key aspect of humans' ability to understand their social world, then both the abilities to select the appropriate knowledge structure and to correctly apply that knowledge are required of a complete cognitive model. If we accept the assumption that this knowledge can be represented by symbolic structures, but reject models which serially examine symbolic data structures, then we need parallel architectures which account for both the predominance of semantic cues in retrieval and the ability to make fine distinctions based on role bindings. We have presented a model here which makes a step towards integrating these two processes using a large network of parallel processors. Furthermore we believe that work of this type is required to develop a stronger version of the physical symbol system hypothesis, a version which not only requires models to be Turing machine equivalent and symbolic in nature, but which also requires models to be real-time (at least potentially), robust, and realizable within the brain.

5. References

Anderson, J. R. (1983). *The Architecture of Cognition.* Harvard University Press.

Bower, G. H., Black, J. B., and Turner, T. J. (1979) Scripts in Memory for Text. *Cognitive Psychology,* 11, 177-220.

Charniak, E. (1986). A Neat Theory of Marker Passing, *in Proceedings of the Fifth National Conference on Artificial Intelligence,* Philadelphia, PA, 584-588.

Cullingford, R. E. (1981). SAM, in R. C. Schank and C. K. Riesbeck (Eds) *Inside Computer Understanding: Five Programs Plus Miniatures.* Lawrence Erlbaum Associates.

DeJong, G. F. (1979). Prediction and Substantiation: A New Approach to Natural Language Processing. *Cognitive Science* 3(3), 251-273.

Dethrick, M. and Plaut, D. C. (1986). Is Distributed Connectionism Compatible with the Physical Symbol System Hypothesis?, in *Proceedings of the Eight Annual Conference of the Cognitive Science Society,* Amherst, MA, 639-644.

Dolan, C. P. and Dyer, M. G. (1987). Symbolic Schemata, Role Binding, and the Evolution of Structure in Connectionist Memories.*Proceeding of the First International Conference on Neural Networks,* San Diego, CA, Volume II, 287-298.

Dolan, C. P. (forthcoming). *The Use and Acquisition of Thematic Knowledge: Symbolic and Connectionist approaches to Learning and Planning.* Ph.D. Dissertation, University of California at Los Angeles, Computer Science Department.

Dyer, M. G. (1983). *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* MIT press.

Fahlman, S. E. (1977). *NETL: A System for Representing and Using Real-World Knowledge.* MIT Press.

Galambos, J. A., Abelson, R. P., and Black, J. B. (1980). *Knowledge Structures.* Lawrence Erlbaum Associates.

Golden, R. M. (1986). Representing Causal Schemata in Connectionist Systems. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society,* Amherst, MA, 13-21.

Hinton, G. E. (1981). Implementing Semantic Networks in Parallel Hardware, in G. E. Hinton and J. A. Anderson (Eds), *Parallel Models of Associative Memory.* Lawrence Erlbaum Associates.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. (1986). Distributed Representation, in D. E. Rumelhart and J. L. McClelland (Eds) *Parallel Distributed Processing,* Volume 1. MIT Press.

Hopfield, J. J. and Tank, D. W. (1985). "Neural" Computation of Decisions in Optimization Problems. *Biological Cybernetics,* 152, 141-152.

Kolodner, J. L. (1983). Maintaining Organization in a Dynamic Long Term Memory. *Cognitive Science,* 7(4), 243-280.

Lebowitz, M. (1983). Generalization from Natural Language Text. *Cognitive Science,* 7(1), 1-40.

McClelland, J. L. and Kawamoto, A. H. (1986). Mechanisms for Sentence Processing: Assigning Roles to Constituents, in J. L. McClelland and D. E. Rumelhart (Eds) *Parallel Distributed Processing,* Volume 2. MIT Press.

Newell, A. (1980). Physical Symbol Systems, *Cognitive Science,* 4, 135-183.

Norman, D. A. (1986). Reflections on Cognition and Parallel Distributed Processing, in J. L. McClelland and D. E. Rumelhart (Eds) *Parallel Distributed Processing,* Volume 2. MIT Press.

Quillian, M. R. (1968). Semantic Memory, in M. Minsky (Ed) *Semantic Information Processing*. MIT Press.

Rumelhart, D. E. and McClelland J. L.(1986). *Parallel Distributed Processing,* Volume 1. MIT Press.

Rumelhart, D. E., McClelland, J. L., and Hinton, G. E. (1986). Schemata and Sequential Thought Processes in PDP Models, in J. L. McClelland and D. E. Rumelhart *Parallel Distributed Processing,* Volume II. MIT Press.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge University Press.

Schank, R. C. (1987). What is AI Anyway?, *AI Magazine.*, 8(4), 57-66.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates.

Seifert, C. M., Dyer, M. G., and Black, J. B. (1986). Thematic Knowledge in Story Understanding. *Text,* 6(4), 393-426.

Shastri, L. and Feldman, J. A. (1985). Evidential Reasoning in Semantic Networks. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence,* Los Angeles, CA, 465-474.

Shastri, L. and Feldman, J. A. (1984). Semantic Networks and Neural Nets. University of Rochester, Computer Science Department Technical Report, TR131.

Smolensky, P. (1986). Information Processing in Dynamical Systems: Foundations of Harmony Theory, in D. E. Rumelhart and J. L. McClelland (Eds) *Parallel Distributed Processing,* Volume I. MIT Press.

Touretzky, D. S., and Hinton, G. E. (1985). Symbols Among the Neurons: Details of a Connectionist Inference Architecture. *Proceedings of Ninth International Joint Conference on Artificial Intelligence,* Los Angeles, CA, 239-243.

Wilensky, R. (1983). *Planning and Understanding*. Addison Wesley.