Computer Science Department Technical Report
Artificial Intelligence Laboratory
University of California
Los Angeles, CA 90024-1596

LEARNING DISTRIBUTED REPRESENTATIONS OF CONCEPTUAL KNOWLEDGE

Geunbae Lee
Margot Flowers
Michael G. Dyer

November 1989
CSD-890061

Learning Distributed Representations of Conceptual Knowledge *†

Geunbae Lee
Margot Flowers
Michael G. Dyer
Artificial Intelligence Laboratory
Computer Science Department, UCLA
gblee@cs.ucla.edu, flowers@cs.ucla.edu, dyer@cs.ucla.edu

Abstract

We argue that distributed representations must satisfy 5 criteria in order to serve as an adequate foundation for constructing and manipulating conceptual knowledge. These criteria are: automaticity, portability, structure encoding, semantic micro-content, and convergence. In our approach, distributed representations of semantic relations (i.e. propositions) are formed by recirculating the hidden layer in recurrent PDP networks. Our experiments show that the resulting distributed semantic representations (DSRs) satisfy all of the above 5 criteria. We believe that DSRs can help supply an important building block in developing more complex connectionist architectures for higher-level inferencing, such as required in natural language processing.

1 Background and Issues

There has been growing concern over how distributed/holographic or localist/punctate representations should be in order to represent high-level knowledge. While Feldman [Feldman, 1986] has given arguments against both extreme punctate and extreme holographic representations, PDP researchers, such as Rumelhart and McClelland [Rumelhart and McClelland, 1986] have listed numerous advantages that distributed representations have over localist representations. At the same time, a number of techniques, e.g. back propagation [Rumelhart *et. al.* , 1986-a] and extended back-propagation [Miikkulainen and Dyer, 1988-a], have been developed for forming distributed representations, including: conjunctive and coarse codings [Hinton *et. al.* , 1986], microfeature based representations [Waltz and Pollack, 1985][McClelland and Kawamoto, 1986], and tensor product representations [Dolan and Smolensky, 1988][Smolenky, 1987].

*This research is supported in part by a contract from the JTF Program of the DoD, monitored by JPL, and by an ITA Foundation Grant. The simulations were carried out on equipment awarded to UCLA by Hewlett Packard.

†Poster presented at IJCNN-89 (Washington D. C.), Abstract appeared in the proceedings.

Developing distributed representations able to support higher-level reasoning and represent conceptual knowledge is not an easy task. Whereas a "von Neumann symbol" starts with random bit string like ASCII code and builds structural relationships between symbols to represent conceptual knowledge, a distributed (or so-called "subsymbolic") representation ought to possess both a structure and a semantics below the symbolic level, namely, as a pattern in an ensemble of neuron-like elements (i.e. creating a "connectionist symbol").

2 Criteria for a Distributed Semantic Representation

A distributed representation able to represent conceptual knowledge must have five features:

(1) *Automaticity* – The representation must be acquired through some automatic learning procedure, rather than set by hand. For instance, the hand-coded microfeature based representation [McClelland and Kawamoto, 1986] does not meet this criterion.

(2) *Portability* – The representation should be global rather than locally confined to its training environment. That is, the representation learned in one training environment should have structural/semantic invariant properties so that it can be applied in another task environment. For example, the representation in Hinton's family tree example [Hinton, 1986] can be said to meet the automaticity criterion, but not the portability criterion, since it cannot be used in any other task.

(3) *Structure Encoding* – Feldman [Feldman, 1986] has argued that any conceptual representation must support answering questions about structural aspects of the concept. For example, part of the meaning of "irresponsible" is that there was an obligation established to perform an action and the obligation was violated. To answer a question about the meaning of "irresponsible" requires accessing these constituent structures. Any conceptual representation must have structural information in the representation itself about the constituents of the concept and purely holographic representations do not meet this criterion. The extended back-propagation method, FGREP[Miikkulainen and Dyer, 1988-a], can be said to meet the first and the second criteria, but the resulting FGREP representation is purely holographic. We can not retrieve any structural information from the representation itself. Thus representations of lexical en-

tries in the FGREP lexicon do not allow us to answer questions about the constituents of any word's conceptual structure. Hand-coded microfeatures are a good representation according to this criterion, since at least one can interpret the semantic content of each microfeature in the representation.

(4) *Micro-Semantics* – Distributed representations gain much of their power by encoding statistical correlations from the training set, which are used to characterize the environment. These statistical correlations give connectionist models the ability to generalize. To support generalization, distributed representations should exhibit semantic content at the micro level, i.e. similar concepts should end up (by some metric) with similar distributed representations. This criterion provided the original impetus for microfeature-based encodings, since similar concepts are similar because they share similar microfeature values.

(5) *Convergence* – A basic operation for any self-organizing (possibly chaotic) representation is convergence to a (possibly chaotic) attractor. At any one time, the representation should have a stable pattern of activation over the ensemble of units in a stable environment, and this pattern should converge to an attractor point in the feature space [Hopfield, 1982].

3 Forming Distributed Semantic Representations (DSRs) of Words

In this section we show how DSRs may be formed and demonstrate their validity for the task of encoding word meanings.

There are two alternate views on the semantic content of words: (1) The structural view defines a word meaning only in terms of its relationships to other meanings. (2) The componential view defines meaning as a vector of properties (e.g. microfeatures). We take an interim view – that meaning can be defined in terms of a distributed representation of structural/functional relationships, where each relationship is encoded as a proposition. Examples of propositions are verbal descriptions of action-oriented events in everyday experiences.

3.1 Representing DSRs

The intuition behind DSRs is that people learn the meanings of words through examples of their relationships to other words. For example, after reading the 4 propositions below, the reader begins to form a hypothesis of what kind of meaning the word "foo" should have.

- Proposition1: The man drinks foo with a straw.
- Proposition2: The company delivers foo in a carton.
- Proposition3: Humans get foo from cows.
- Proposition4: The man eats bread with foo.

The meaning of foo should be something like that of "milk". The interesting fact is that the semantics of "foo" is not fixed, rather it is gradually refined as one experiences more propositions in varying environments. To develop DSRs based on propositions, we have to define the structural/functional relationships between concepts with respect to those propositions. For action-oriented events describing propositions, we use thematic case relations, originally developed by Fillmore [Fillmore, 1968], and extended in several natural language processing systems [Schank and Riesbeck, 1981]. We use the following 8 thematic case relations which are similar to the ones defined in [Fillmore, 1968]: agent, object, co-object, instrument, source, goal, location, and time. For example, the DSR of "milk" is now defined as the composition of relationships, e.g. with respect to these 4 Propositions:

> *milk* = F_i (G_c (Object, *Proposition1*), G_c (Object, *Proposition2*), G_c (Object, *Proposition3*), G_c (Co-object, *Proposition4*),.....)

where *milk* is the meaning representation of "milk"; F_i is some integration function and G_c is some combination function of structural/functional relationships with respect to the corresponding propositions. In the same way, each proposition is temporally defined as the composition of the constituent thematic case components that are themselves combinations of structural/functional relationships with their corresponding meaning representations of words:

> *proposition1* = F_i (G_c (agent, *man*), G_c (verb, *drink*), G_c (object, *milk*), G_c (instrument, *straw*))

3.2 Learning DSRs

We have developed auto-associative recurrent PDP (ARPDP) networks for automatically learning DSRs. The basic idea is to "re-circulate" the developing internal representation (hidden layer of the network) back out to the environment (input and output layers of the network). This idea has been suggested by various researchers, e.g. FGREP [Miikkulainen and Dyer, 1988-a][Miikkulainen and Dyer, 1988-b], Recursively Reduced Descriptions [Hinton *et. al.* , 1986], Recursive Auto-Associative Memories [Pollack, 1988], Sequential Connectionist Networks [Jordan, 1986][Elman, 1988], and has been used in natural language question-answering [Allen, 1988], parsing [Hanson and Kegl, 1987] and sentence comprehension [St. John and McClelland, 1988].

Figure 1 shows our system architecture, ARP.

The learning portion of the ARP architecture contains two symbolic memories (Concexicon and Proposition buffers) and two 3-layer ARPDP networks. The input and output layers of each network has 3 banks of units: bank1, bank2, bank3. After each of the 3 banks is properly loaded, the DSR emerges in bank1 by unsupervised auto-associative BEP (Backward Error Propagation) [Rumelhart *et. al.* , 1986-a].

The DSR learning process consists of two alternating cycles: Concept Encoding and Proposition Encoding. Below we informally describe each cycle. In each, all concept and proposition representations start with a DON'T CARE pattern, e.g. 0.5, when the activation value range of each unit in network is 0.0 to 1.0. The structural/functional relationship representation is fixed using orthogonal bit patterns (for minimizing interference).

Concept Encoding Cycle:

1. Pick one concept to be represented, say CON1.

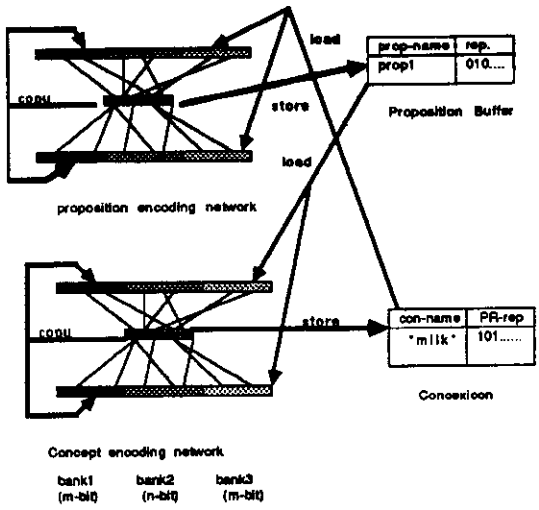

Figure 1: ARP Network Architecture for Learning DSRs

2. Select all relevant triples for CON1. In the *milk* example, they should be triples like (*milk* object proposition1) (*milk* object proposition2) (*milk* object proposition3), etc.
3. For the first triple, load the initial representation for CON1 into bank1; load the structural/functional relationship into bank2, and load its corresponding proposition to bank3. In the *milk* example, for the first triple, bank1, bank2, and bank3 are loaded with bit patterns for *milk*, object, proposition1 respectively.
4. Run the auto-associative BEP algorithm, where the input and output layers have the same bit patterns.
5. Re-circulate the developed (hidden layer) representation into bank1 of both the input/output layers and perform step3 to step5 for another triple until all triples are consumed.
6. Store the developed DSR into the concexicon and select another word concept to be represented.

Proposition Encoding Cycle: Basically this cycle undergoes the same steps as the Concept Encoding Cycle except that, this time, we load bank1, bank2, and bank3 with (respectively) the proposition to be represented, structural/functional r elationship, and its corresponding concept representation (DSR). The result of the encoding is stored into the proposition buffer which can be flushed and reused after we acquire all the necessary stable bit patterns for all concepts.

Now the overall DSR learning process will be:

1. Perform the entire concept encoding cycle.
2. Perform the entire proposition encoding cycle.
3. Repeat step1 and step2 until we get stable patterns for all concepts.

In this process,the composition function F_i is embodied in the dynamics of the Recursive Auto-Associative Stacking operation [Pollack, 1988] and the combination function G_e is just a concatenation of two bit patterns.

4 Evaluation of Distributed Semantic Representations

Does the learned distributed semantic representations meet the aforementioned 5 criteria? It might be intuitively clear that DSR satisfies criteria (1) through (3). Demonstrating satisfaction of criteria (4) and (5) is not as easy since semantic micro-content and convergence both depend on the learning environment (i.e. the propositions chosen).

We must consider two important conditions needed to make DSRs work correctly: (a) Selected propositions should reflect the real protocols by which people acquire a given word meaning or other concept. (b) The defined structural/functional relationships should provide the basic building blocks for word semantics[Schank, 1973].

The DSR approach satisfies the five criteria:

(1) *Automaticity:* DSR is automatically learned by using ARPDP networks, rather than set by hand.

(2) *Portability:* Since each DSR is learned without any dependence on any particular task, its encoded propositional contents can be ported to any application environment. As a result, the representation has structural and semantic properties that are invariant over all task environments.

(3) *Structure Encoding:* Each DSR was learned by stacking the structural/functional relationship and proposition pairs. These propositions again can be decoded to return the constituent relationships and concepts. Therefore the representation itself supports the answering of structural questions about concept.

The decoding process [Pollack, 1988] is the reverse process of encoding: We load the concept representation in the hidden layer of the ARP concept encoding network and perform relaxation until we get the desired relationship in bank2 and proposition in bank3 of the output layer. Next, we load the resulting proposition in the hidden layer of the proposition encoding network and get back the constituent relationships and concept representations.

Figure 2 shows our decoding architecture.

(4) *Semantic Micro-Content:* Our rationale for this criteria is that similar concepts should function as similar case roles for the similar propositions, so they should develop similar distributed semantic representations when starting from all same DON'T CARE patterns. For example, the concept of *milk* functions in a similar case-role to the concept of *juice* in the INGESTing type propositions [Schank, 1973] than the concept of, say, *man*. So *milk* and *juice* will end up acquiring more similar distributed semantic representations.

(5) *Convergence:* This criterion depends on operational parameters such as learning rate, momentum factor, number of training epochs etc. in BEP network learning algorithms. (See [Fahlman, 1988] for an empirical study.) We have demonstrated convergence experimentally.

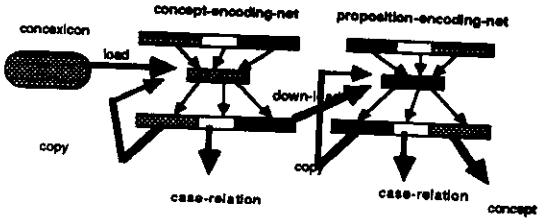

Figure 2: ARP Decoding Architecture

5 Experiment: Learning DSRs for Nouns and Verbs

We conducted a number of experiments to see how well ARPDP networks learned DSRs for nouns and verbs. We made up over 100 propositions and analyzed their case structures in order to load them into our network architecture, ARP.

Figure 3 shows the case structure of propositions used.

Figure 4 shows the DSRs learned for a number of nouns and verbs. The learned representations exhibit semantic micro-content properties according to the Concept categories. The result is a snapshot after 30 epochs. By this point the representation has stabilized and converged to certain attractor points. With excessive simulation epochs, the representations begin to reflect the minute details and statistical biases of the simulation data.

Figure 5 shows the similarity structure of the learned DSRs in terms of their Euclidean distances. They form clusters according to the concept categories along the main diagonal.

Interestingly enough, the representation of each proposition exhibits also the similarity structures. Figure 6 shows parts of their representations. This representation is a gestalt representation for each event and could be used in a connectionist schema processing system like reported in [Chun and Mimo, 1987].

DSRs show many similar characteristics to those reported in [Miikkulainen and Dyer, 1988-a][Miikkulainen and Dyer, 1988-b], but unlike FGREP representations, DSRs appear to be more portable based on their encoding of propositional content. Each DSR can also reconstruct its constituent information through the decoding process. Moreover, DSRs are learned independent of any particular processing task, so the representations should be useful in any task requiring access of the propositional content of word meanings.

6 Decoding DSRs into Their Constituents

We also conducted decoding experiments to demonstrate the representation's structure encoding property (using

p#	proposition	case-structure
1	man ate	AV
3	man ate chicken	AVO
9	man ate chicken with fork	AVOI
19	bat ate	AV
23	man ate chicken at home	AVOL
35	man drank	AV
37	man drank milk	AVO
41	man drank milk at home	AVOL
49	dog drank	AV
51	man drank milk with straw	AVOI
55	man broke plate	AVO
59	man broke plate with ball	AVOI
71	ball broke plate	IVO
77	bat broke plate	AVO
85	plate broke	OV
87	man moved	AV
89	man moved ball	AVO
98	man moved ball from home	AVOS
107	man moved ball to home	AVOG
116	bat moved	AV
120	ball moved	OV

Figure 3: Proposition Types and their Case Structures. Note that the proposition number is not contiguous. It shows only parts of the propositions actually used. (A:agent V:verb O:object I:instrument L:location S:source G:goal)

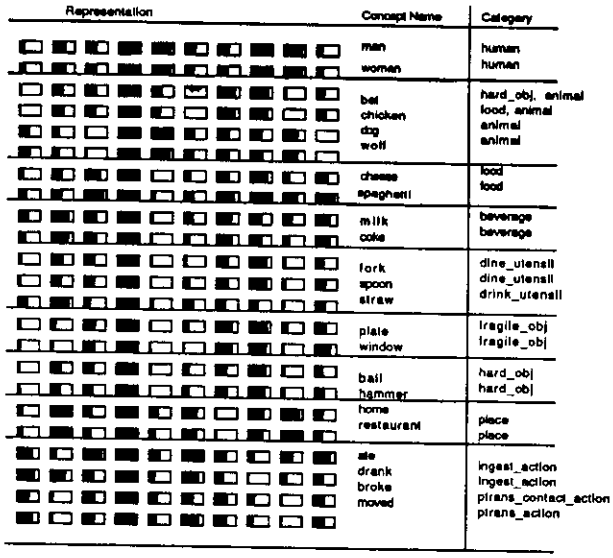

Figure 4: Learned DSRs of Nouns/Verbs with their Concept Category. The experiment is done using momentum accelerated backpropagation. Learning rate = 0.07; Momentum factor = 0.5; 30 epochs for each concept; one epoch = 100 cycles of auto-associative backpropagation. The value range is 0.0-1.0 continuous which is shown by the degree of box fill-up.

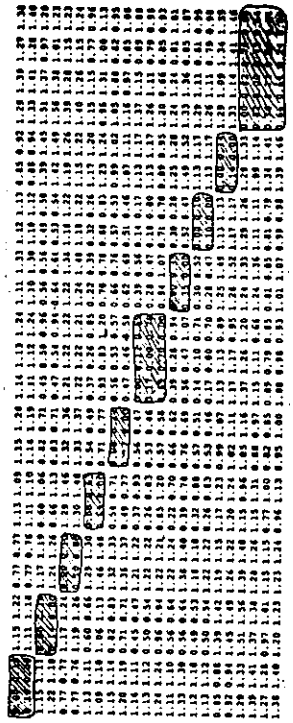

Figure 5: The Euclidean Distance 2 by 2 Matrix between each learned DSR of nouns and verbs. Each row/column designates the 23 values for: man, woman, bat, chicken, dog, wolf, cheese, spaghetti, milk, coke, fork, spoon, straw, plate, window, ball, hammer, home, restaurant, ate, drank, broke, moved (in this order).

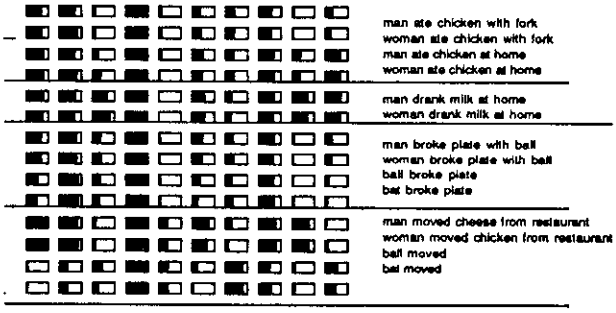

Figure 6: Learned Representations for Propositions (Events). Learned under the same conditions as in Figure 4.

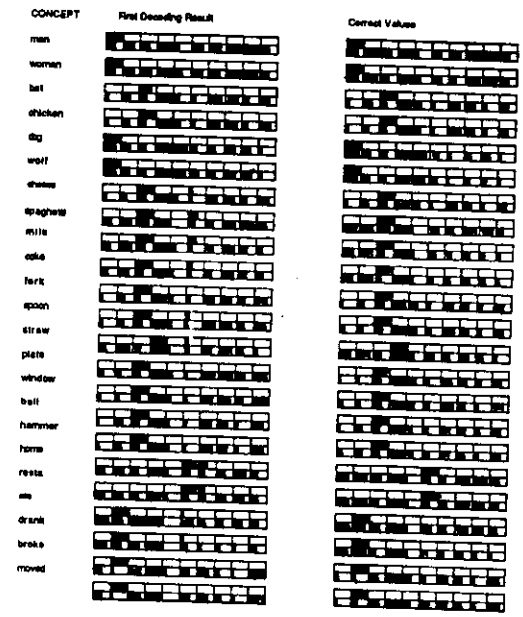

Figure 7: Decoding Result for Concept Representations. For each concept, the first row designates the orthogonal fixed representation of case-relations (bank2), and the adjacent second row designates the representation for the constituent propositions (bank3). The first column shows the first decoded result (stack-top) and the second column shows their corresponding correct values, for comparison.

the architecture shown in Figure 2). Figure 7 shows the decoding results. Since we can think of each DSR as a stack of (case-relation, proposition) pairs, the decoding operation is like stack-popping operation. We get constituent pairs in a Last-In-First-Out (LIFO) fashion.

The decoding performance is good, as can be seen in Figure 7. Each concept representation has been demonstrated to contain its own structural information, such as constituent case-relation and proposition pairs.

Figure 8 shows the decoding result for Proposition Representations in the same way.

7 Current Status

The eventual objective of this work is to develop distributed knowledge representations that can be utilized in high-level reasoning systems. Just as the von Neumann symbolic representation is utilized as a building block in symbolic AI systems, we want to use DSRs as a building block in connectionist or connectionist/symbolic hybrid models [Dyer, 1988] able to support such tasks as natural language processing. One example is a connectionist schema processing system. The problem of previous connectionist schema processing systems

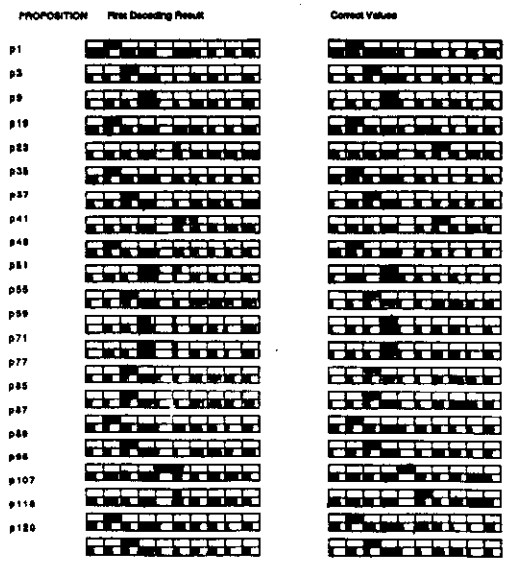

Figure 8: Decoding Result for Proposition Representations. The proposition number corresponds to the number shown in Figure 3. For each proposition, the first row designates the case-relation representation and the second row designates the constituent concept representation.

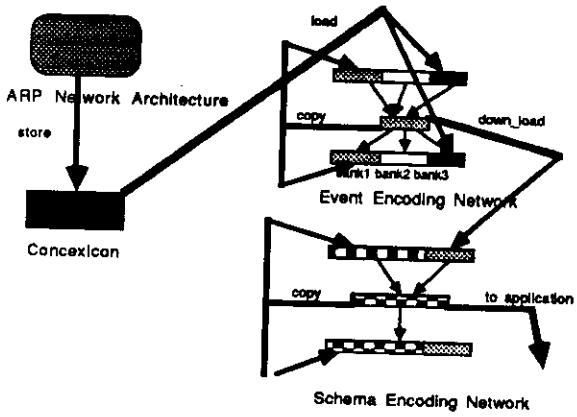

Figure 9: Connectionist Schema Processing model using Learned DSRs

[Rumelhart *et. al.* , 1986-b][Chun and Mimo, 1987] is that, while they have nodes for object/events, they do not have any underlying semantic micro-representations for those nodes.

DSRs can be used as a basic concept representation scheme which can be integrated into event representations, as shown in Figure 9. We are currently performing experiments on application of DSRs to connectionist schema processing. The use of DSRs for event/object representations will create representations with semantic micro-content and therefore exhibit more generalization and fault-tolerance properties.

8 Conclusion

We have discussed 5 criteria that distributed representations must exhibit if they are to serve as building blocks for higher-level knowledge processing tasks. Our distributed semantic representations (DSRs) have been implemented using auto-associative recurrent PDP networks in an architecture called ARP and experiments have shown that DSRs meet all 5 criteria. The next step is to use DSRs as building blocks in a more complex, high-level connectionist reasoning systems.

References

[Allen, 1988] Allen, Robert B. Sequential Connectionist Networks for Answering Simple Questions about a Micro-World. *Proc. of the Tenth Annual Conference of the Cognitive Science Society,* Montreal, August 1988.

[Chun and Mimo, 1987] Chun, Hon Wai and Alejandro Mimo. A model of schema selection using marker passing and connectionist spreading activation. *Proc. of the Ninth Annual Conference of the Cognitive Science Society,* Seattle, WA, 1987.

[Dolan and Smolensky, 1988] Dolan, Charles P. and Smolensky Paul. Implementing a Connectionist Production System Using Tensor Products. *Proc. of the 1988 Connectionist Model Summer School,* Morgan Kaufmann, 1988.

[Dyer, 1988] Dyer, M.G. Symbolic NeuroEngineering for Natural Language Processing: a Multilevel Research Approach. Technical Report UCLA-AI-88-14 and also in J. Barnden and J. Pollack (Eds) *Advances in Connectionist and Neural Computation Theory.* Ablex Publ., in press.

[Elman, 1988] Elman, J.L. Finding Structure in Time. Technical Report 8801, Center for Reseach in Language. UCSD, San Diego. 1988.

[Fahlman, 1988] Fahlman, Scott E. An Empirical Study of Learning Speed in Back-Propagation Networks. Tech Report CMU-CS-88-162, CS Dept., Carnegie Mellon Univ., June, 1988.

[Feldman, 1986] Feldman, J. A. Neural Representation of Conceptual Knowledge. Technical Report TR189, Dept. of CS, Univ. of Rochester, NY.1986.

[Fillmore, 1968] Fillmore, C. The Case for Case. In E. Bach and R. Harms (Eds). *Univerals in linguistic theory,* NewYork: Holt, Rinehart and Winton 1968.

[Hanson and Kegl, 1987] Hanson, Stephen J. and Kegl, Judy. PARSNIP: A Connectionist Network that Learns Natural Language Grammer from Exposure to Natural Language Sentences. *Proc. of the Ninth Annual Conference of the Cognitive Science Society,* Seattle, WA, 1987.

[Hinton *et. al.* , 1986] Hinton, G. E., McClelland, J. L. and Rumelhart, D. E. Distributed Representations.

In Rumelhart and McClelland. *Parallel Distributed Processing,* Vol 1, Bradford Book/MIT Press, 1986.

[Hinton, 1986] Hinton, G. E. Learning Distributed Representation of Concepts. *Proc. of the Eighth Annual Conference of the Cognitive Science Society,* Amherst, MA, 1986.

[Hopfield, 1982] Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. of National Academy of Science,* Vol 79, pp 2554-2558, 1982.

[St. John and McClelland, 1988] St. John, M.F. and McClelland, J. L. Applying contextual constraints in sentence comprehension. *Proc. of the Tenth Annual Conference of the Cognitive Science Society,* Montreal, August, 1988.

[Jordan, 1986] Jordan, M.I. Serial Order: A parallel distributed processing approach. Tech Report 8604. Institute for Cognitive Science. UCSD, San Diego. 1986.

[McClelland and Kawamoto, 1986] McClelland, J. L. and Kawamoto, A. H. Mechanisms of sentence processing: assigning roles to constituents of sentences. In McClelland and Rumelhart (Eds.) *Parallel Distributed Processing,* Vol 2, Bradford Book/MIT Press 1986.

[Miikkulainen and Dyer, 1988-a] Miikkulainen, R. and Dyer, M.G. Forming Global Representation with Extended BackPropagation. *Proc. of the IEEE Second Annual International Conference on Neural Nets (ICNN-88),* San Diego, CA. July 1988.

[Miikkulainen and Dyer, 1988-b] Miikkulainen, R. and Dyer, M.G. Encoding Input/Output Representations in Connectionist Cognitive Systems. *Proc. of the 1988 Connectionist Model Summer School,* Morgan Kaufmann, 1988.

[Pollack, 1988] Pollack, J. Recursive Auto-Associative Memory: Devising Compositional Distributed Representations. *Proc. of the Tenth Annual Conference of the Cognitive Science Society,* Montreal, 1988.

[Rumelhart and McClelland, 1986] Rumelhart, D. E. and McClelland, J. L.(Eds.) *Parallel Distributed Processing: Explorations into the microstructure of cognition* (Vols. 1 and 2) Bradford Book/MIT Press, 1986.

[Rumelhart *et. al.* , 1986-a] Rumelhart, D. E., Hinton, G. E. and Williams, R. Learning Internal Representations by Error Propagation. In Rumelhart and McClelland (Eds.) *Parallel Distributed Processing,* Vol. 1, 1986.

[Rumelhart *et. al.* , 1986-b] Rumelhart,D.E., Smolensky, P., McClelland, J. L. and Hinton, G. E. Schemata and Sequential Thought Processes in PDP Models. In Rumelhart and McClelland (Eds.) *Parallel Distributed Processing,* Vol. 2, 1986.

[Schank, 1973] Schank, R. C. Identification of conceptualization underlying natural language. In Schank and Colby (Eds). *Computer models of thought and language,* W.H. Freeman and Company, 1973.

[Schank and Riesbeck, 1981] Schank, R.C. and Riesbeck, C.K. *Inside Computer Understanding,* Lawrence Erlbaum Assoc., Hillsdale NJ 1981.

[Smolenky, 1987] Smolensky, Paul. A method for connectionist variable binding. Tech Report CU-CS-356-87, CS Dept and Institute of Cognitive Science, Univ. of Colorado, Feb 1987.

[Waltz and Pollack, 1985] Waltz, D. L. and Pollack, J. B. Massively parallel parsing: A strong interactive model of natural language interpretation. *Cognitive Science,* Vol. 9, pp51-74, 1985.