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

HYBRID CONNECTIONIST MODELS: TEMPORARY BRIDGES OVER THE GAP BETWEEN THE SYMBOLIC AND THE SUBSYMBOLIC

Trent E. Lange

April 1991
CSD-910011

Hybrid Connectionist Models: Temporary Bridges Over the Gap Between the Symbolic and the Subsymbolic

Trent E. Lange

April 1991

Technical Report UCLA-AI-91-04

Hybrid Connectionist Models: Temporary Bridges Over the Gap Between the Symbolic and the Subsymbolic†*

Trent E. Lange

Artificial Intelligence Laboratory
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90024

ABSTRACT

Connectionist models can be broken into three major levels, ranging from the subsymbolic to the symbolic. These are: (1) *distributed connectionist networks,* commonly referred to as subsymbolic models, which represent knowledge as distributed patterns of activation across simple numeric processing elements, (2) *structured connectionist networks,* which also use simple numeric processing elements, but which are structured according to knowledge representations similar to symbolic models, and (3) *marker-passing networks,* which are symbolic models that utilize the massive-parallelism of connectionism. Because of their associated constraints and abilities, each level of connectionist modelling offers advantages and disadvantages in constructing models for understanding the processes of cognition.

While staying true to one connectionist approach has its appeal, the current limitations of each level often restricts the tasks one is able to perform. For researchers interested in modelling a particular human capability, it is therefore sometimes necessary to build *hybrid* models that are built of elements from more than one level of connectionist modelling. In this paper it is argued that building such hybrid models supports the long-term goal of mapping high-level cognitive functions into the neural level of the brain by allowing progress on levels that might otherwise be stymied and by highlighting areas that need more extensive research. To illustrate the benefits of this approach, this paper describes three models: (1) a hybrid network that is able to model an integration of cognitive functions not easily plausible in a single network level, (2) a structured connectionist network that illustrates that the abilities of hybrid models can often eventually be built into a single network level, and (3) a distributed connectionist network that is a step towards removing one of the *hidden* hybrid mechanisms found in most connectionist models.

---

†This paper will appear as a chapter in J. Dinsmore (ed.), *Closing the Gap: Symbolic vs. Subsymbolic Processing*. Hillsdale, NJ: Lawrence Erlbaum Associates.

*This research has been supported in part by grants from the W.M. Keck and ITA Foundations. I would like to thank John Dinsmore and Michael Dyer for their helpful comments and for the opportunity to present the arguments and models in this paper. I would also like to acknowledge the contributions of Keith Holyoak, Eric Melz, Charles Wharton, and Colin Allen to the development of the models described here and for their valuable suggestions on earlier drafts of this chapter.

1. INTRODUCTION AND MOTIVATION

Connectionist networks, often known as neural networks or spreading-activation networks, have recently been the subject of a tremendous rebirth of interest, as researchers have begun to explore their advantages for cognitive models ranging from low-level sensory abilities to high-level reasoning. Connectionist models employ *massively parallel* networks of relatively simple processing elements that draw their inspiration from neurons and neurobiology, as opposed to traditional symbolic artificial intelligence (AI) models, which are generally based on serial Von Neumann architectures. Within the connectionist paradigm there are three major levels: *distributed connectionist networks* and *localist connectionist networks*, and *marker-passing networks*, each having different types of cognitive models that they are best suited for[1].

Distributed connectionist networks, sometimes known as *Parallel Distributed Processing* or *subsymbolic* models, are networks which represent knowledge as *distributed* patterns of activation across their units (see [Dinsmore, this volume]). Most distributed network models have learning rules (such as backpropagation [Rumelhart, Hinton, & McClelland, 1986]), to *train* their connections' weights to generate desired input/output behavior. With such training rules, distributed networks are able to perform statistical category generalization, perform noise-resistant associative retrieval, and exhibit robustness to damage. They have been successfully employed for low-level tasks such as visual pattern recognition [Fukushima, Miyake, & Ito, 1983], speech consonant recognition [Waibel, 1989], and assigning roles to constituents of sentences [McClelland & Kawamoto, 1986]. On the other hand, distributed networks have had difficulty with both dynamic variable bindings and the representation of structure needed to handle complex conceptual relationships, and so are not currently well-suited for high-level cognitive tasks such as natural language understanding and planning.

Localist connectionist networks, sometimes known as *structured* or *spreading-activation* networks, also use units with simple numeric activation and output functions (see [Dinsmore, this volume]), but instead represent knowledge using *semantic networks* in which concepts are represented by individual units and their labelled interconnections. Unlike distributed networks, localist networks are parallel at the knowledge level and have structural relationships between concepts built into the connectivity of the network. Because of this, localist networks are especially well-suited for cognitive tasks such as word-sense disambiguation [Waltz & Pollack, 1985], limited inference [Shastri, 1988], and language generation [Gasser, 1988]. Unfortunately, localist networks lack the powerful learning and generalization capabilities of distributed networks and also have had difficulty with dynamic variable bindings and other capabilities of symbolic models.

Marker-passing networks are unlike distributed networks and localist networks in that their units do not use numeric activation functions, but instead use built-in symbolic capabilities. Like localist networks, they also represent knowledge in semantic networks and retain parallelism at the knowledge level. However, instead of spreading numeric activation values, marker-passing networks propagate symbolic markers, and so support the variable binding necessary for rule application while retaining the power of symbolic systems. Because of this, they have been able to approach high-level areas such as planning [Hendler, 1988] and natural language understanding [Charniak, 1986]. On the downside, marker-passing networks' units are

[1]There is some disagreement as to whether marker-passing models should be classified as connectionist models, given their explicit use of symbolic processing. In this chapter, we will consider marker-passing models to be connectionist (in a broad sense) to emphasize their shared processing philosophies of massively-parallel and relatively simple (compared to traditional symbolic models) processing elements.

more complex than those of distributed networks and localist networks, they do not possess the learning capabilities of distributed networks, and they do not exhibit the constraint-satisfaction capabilities of localist networks.

Most connectionist researchers have explored and built models within a single connectionist level. However, while staying true to one connectionist approach has its appeal, the current limitations of each level often restricts the tasks one is able to perform. For researchers interested in modelling a particular human capability, it is therefore sometimes necessary to build *hybrid* models that are built of elements from more than one level of connectionist modelling[2]. In this paper it is argued that building such hybrid models supports the long-term goal of mapping high-level cognitive functions into the neural level of the brain by allowing progress on levels that might otherwise be stymied and by highlighting areas that need more extensive research.

To illustrate the benefits of this approach, this paper describes three models: (1) a localist connectionist network that illustrates that the abilities of hybrid models can often eventually be built into a single network level, (2) a hybrid network that is able to model an integration of cognitive functions not easily plausible in a single network level, and (3) a distributed connectionist network that is a step towards removing one of the *hidden* hybrid mechanisms found in most connectionist models.

1.1. Connectionist and Symbolic Models

To understand the need to build hybrid models, it is important to know the abilities and limitations of each connectionist level. While it is possible that a single type of connectionist model (such as distributed connectionist networks) will eventually be able to model all levels of human cognition, this is certainly far from the case now. However, models from all levels of connectionist networks taken together currently span a wide range of human abilities (if only to a limited depth), ranging from low-level perceptual tasks to high-level reasoning.

One area that has been approached by all types of connectionist and symbolic models is that of semantic natural language understanding. Natural language understanding is a good area to illustrate the benefits and drawbacks of each connectionist level, since it requires a whole range of abilities ranging from low-level pattern matching (such as retrieval of word meanings and simple case-role filling), to high-level manipulation of complex symbolic representations (such as for comprehending intricate stories or editorials), to working with noisy and incomplete data (requiring disambiguation and reinterpretation), to learning and generalization.

1.1.1. Symbolic Rule-Based Models

Symbolic artificial intelligence (AI) systems have so far been the types of models best able to perform high-level reasoning and natural language understanding. A good example is BORIS [Dyer, 1983], a natural language understanding program for modelling in-depth understanding of relatively long and complex sto-

---

[2]The term "hybrid model" is sometimes used to refer to networks that attempt to explicitly duplicate symbolic processing abilities. Marker-passing models are often called hybrid or symbolic models because of their explicit propagation of symbolic markers. The term is also occasionally used to refer to normal localist networks, since their units have symbolic labels (though the labels generally do not affect network processing). In this chapter, however, the term "hybrid model" is used only to describe models that combine elements from more than one level (distributed, localist, marker-passing) of connectionist processing.

ries. BORIS had a hand-coded symbolic knowledge base containing knowledge structures representing various actions, plans, goals, emotional affects, and methods for avoiding planning failures. When reading in a story, BORIS would fire rules from its knowledge base to perform inferencing and form an internal representation of the story, about which it could then answer questions. Other models that have successfully approached complex parts of the language understanding process have all had similar types of knowledge representation and rule-firing capabilities.

Connectionist networks, however, have significant potential advantages over traditional symbolic approaches to the interpretation process. Their conceptual knowledge is stored entirely in an interconnected network of units whose states are computed in parallel, calculated solely by local update functions that are based on their previous state and that of the units to which they are connected. As a result, a major portion of the understanding process could potentially be controlled by a relatively simple and local spreading-activation mechanism, instead of by large collections of brittle and sometimes ad-hoc rules.

1.1.2. Distributed Connectionist Networks

Distributed connectionist models have had a great deal of success modelling low-level natural language understanding tasks, especially those requiring similarity-based learning. A number of researchers have argued that this new subsymbolic paradigm will completely subsume the symbolic paradigm, as the explicit rules used in symbolic models are replaced by the more robust interactions of distributed representations and connection weights learned from experience [Rumelhart & McClelland, 1986]. Although some of the severest criticisms of this stand ([Fodor & Pylyshyn, 1988], [Pinker & Prince, 1988]) have been partially rebutted by recent models showing that distributed models can represent some limited variable bindings and constituent structure (e.g. [Touretzky & Hinton, 1988], [Pollack, in press]), current distributed models are still quite limited in comparison to symbolic models in their abilities to perform high-level processing such as natural language understanding.

A good example of how distributed connectionist models have been used to approach language understanding is provided by the case-role assignment model of McClelland & Kawamoto [1986]. The main task of their model is to learn to assign the proper semantic case roles for sentences. For example, given the syntactic surface form of the sentence *The boy broke the window*, their network is trained to place the semantic microfeature representation of the subject Boy into the units representing the Agent role on the output layer, whereas given *The rock broke the window*, it is trained to place the representation of the subject Rock into the Instrument role. Their network is also trained to perform lexical disambiguation, e.g. mapping the pattern for the word *bat* to a Baseball-Bat for sentences such as *The boy hit the ball with the bat*, and to a Flying-Bat for sentences such as *The bat flew*. Once the input/output pairs have been learned, the network exhibits a certain amount of generalization by mapping the case roles and performing lexical disambiguation for novel inputs similar to the training sentences.

One of the main limitations of McClelland & Kawamato's model for language understanding is that its output can only handle direct, one-step mappings from the input to the output. This limits it to processing isolated sentences, and only those that can be understood and disambiguated based upon the surface semantics of the input alone. Two distributed connectionist models that get around this limitation are the models of Miikkulainen & Dyer [1989] and St. John [1990]. Both models use *recurrent networks* with a hidden layer of units whose activation pattern essentially stores the state (or "gestalt") of the stories being understood. This allows them to learn to process more complex language based on scripts (such as going to a restaurant) and other script-like stories [Schank & Abelson, 1977]. Both models have the lexical disambiguation abilities of McClelland & Kawamoto's model, but, more importantly, are able to infer unmentioned story events and role-fillers from the script that has been recognized by the hidden layer.

Unfortunately, there may be significant problems in scaling such *pattern-transformation* distributed connectionist models to handle more complex language. Both Miikkulainen & Dyer and St. John's models work by resolving constraints from input context to recognize one of their trained scripts and instantiate it with the bindings of the particular input story. However, much of language understanding involves the inference of causal relationships between events for completely novel stories in which no script or previously-trained input/output pair can be recognized. This requires *dynamic inferencing* — a process of constructing chains of inferences over simple known rules, with each inference resulting in a potentially novel intermediate state [Touretzky, 1990]. It remains to be seen whether a single blended activation pattern on the bank of hidden units in recurrent networks can simultaneously hold and make dynamic inferences from multiple, never-before encountered interpretation chains.

Other distributed models explicitly encode variables and rules, such as the models of Touretzky & Hinton [1988] and Dolan & Smolensky [1989]. Because of this, such *rule-implementing* distributed models are able to perform some of the dynamic inferencing necessary for language understanding. Unfortunately, however, the types of rules they can currently encode are generally limited. More importantly, they are serial at the knowledge level because they can fire only one rule at a time. This is a serious drawback for natural language understanding, particularly for ambiguous text, in which the often large number of multiple alternative interpretations often requires that the inference paths be explored in parallel [Lange, in press].

1.1.3. Localist Connectionist Networks

Localist connectionist models represent knowledge in semantic networks in which concepts are represented by individual units and relations between concepts are encoded by weighted connections between those units. The numeric activation level on each conceptual unit generally represents the amount of *evidence* available for its concept in a given context. Because knowledge is spread across the network (as opposed to the concentration of knowledge in the weights between the single input and output layer of most distributed models), localist models have the potential to pursue multiple candidate interpretations of a story in parallel as each interpretation is represented by activation in different local areas of the network. This makes them ideally suited to the disambiguation portion of the language understanding process, because it is achieved automatically as related concepts under consideration provide graded activation evidence and feedback to one another in a form of analog constraint relaxation.

As an example of how localist connectionist models process language and perform disambiguation, consider the sentence:

The astronomer married the star. (**Star-Marriage**)

The word *star* could be easily disambiguated to Movie-Star by a symbolic rule-based system having selectional restrictions (even astronomers cannot marry celestial bodies, except perhaps metaphorically). However, many readers report this and similar sentences as "cognitive doubletakes" because *astronomer* initially primes the Celestial-Body interpretation. Figure 1 shows an extended version of the semantic portion of the localist network Waltz & Pollack [1985] built to process **Star-Marriage** and illustrate this effect. After the input units for **Star-Marriage** are clamped to a high level of activation, the Celestial-Body interpretation of *star* initially acquires more activation than the Movie-Star interpretation because of priming from Astronomer through Astronomy (Figure 2). However, Movie-Star eventually wins out because activation feedback over the semantic connections from the Marry unit to Movie-Star outweighs that spreading from the Astronomer unit to Celestial-Body.

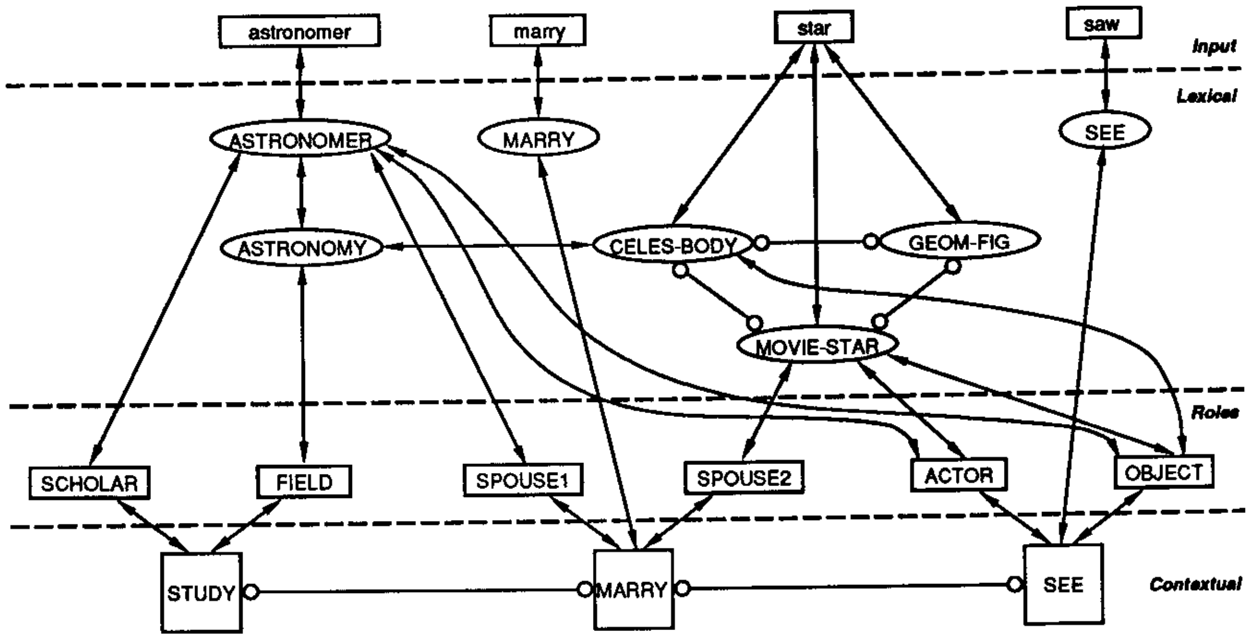

Figure 1. Localist spreading-activation network based on [Waltz & Pollack, 1985]. Lines with arrows are excitatory connections; lines with open circles are inhibitory.

Unfortunately, the applicability of localist connectionist models to natural language understanding has been severely hampered because of their difficulties representing dynamic role-bindings and performing inferencing[3]. Their lack of variable binding abilities leaves them prone to crosstalk even for simple sentences. For example, the network of Figure 1 has no way to distinguish between the sentences *The astronomer saw the star* and *The star saw the astronomer,* despite the crucial difference that the role-bindings make in their interpretation. More importantly, without a mechanism to represent such dynamic bindings, they cannot propagate them to make the chains of inferences necessary for understanding more complex language. This has so far stopped them from going beyond simple language processing that can be resolved based solely on the surface semantics of the input.

1.1.4. Marker-Passing Networks

Marker-passing models operate by spreading symbolic markers in parallel across labelled semantic networks similar to those of localist connectionist networks. Interpretation of the input is achieved when propagation of markers finds a path of units connecting words and concepts from the input text. Because of the symbolic information held in their markers and networks, they are able to represent dynamic role-bindings, and so have been able to perform high-level inferencing for natural language understanding (cf. [Charniak, 1986], [Riesbeck & Martin, 1986], [Granger, Eiselt, & Holbrook, 1986], [Eiselt, 1987], and [Norvig, 1989]).

---

[3]Ajjanagadde & Shastri [1989], Barnden [1990], and Holldobler [1990] describe structured models that can perform some variable-binding and inferencing, but which do not have the disambiguation abilities of normal structured spreading-activation models.

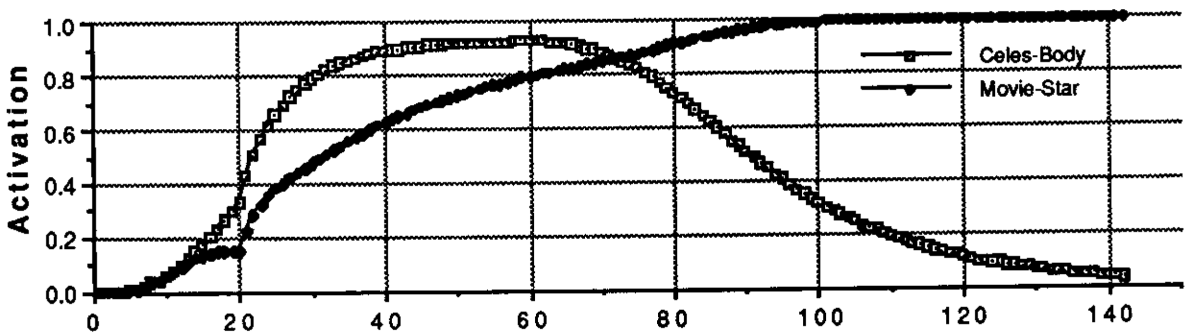

Figure 2. Activations of meaning of word *star* after *astronomer married star* is clamped for network in Figure 1.

As an example of how marker-passing networks process language and perform disambiguation, consider the following text (from [Eiselt, 1987]):

Fred asked Wilma to marry him. Wilma began to cry. **(Marriage)**

Interpretating this text requires that a causal relationship be inferred between Fred's proposal and Wilma's crying. One possible reason for her crying was that she was happy about his proposal and crying "tears of joy". To understand this sentence and resolve the ambiguity, ATLAST [Eiselt, 1987] uses the network shown in Figure 3 by passing markers starting from the units for Cry-Tears and Propose-Marriage. This propagation of markers finds the path Cry-Tears ↔ Happy-State ↔ Happy-Event ↔ Propose-Marriage, returning the "tears of joy" interpretation. Besides finding the inference path representing the interpretation of the story, the symbolic pointers held in the markers also keep track of the role-bindings, so that the model can clearly resolve that it was Fred who did the Propose-Marriage and Wilma who did the Cry-Tears, and not the other way around.

Much text, of course, is ambiguous and has multiple possible interpretations, and the **Marriage** example is no exception. Another possible reason that Mary began to cry was that she was saddened or upset by Fred's proposal. The same propagation of markers that found the above "tears of joy" path will therefore find a second path, Cry-Tears ↔ Sad-State ↔ Sad-Event ↔ Propose-Marriage. To resolve such ambiguities, marker-passing systems generally use a serial heuristic path evaluator separate from the marker-passing process to select the most relevant path from the many paths generated. Such path evaluators usually include rules that select shorter paths over longer ones, reject paths that do not include as much of the input as competing ones, and so forth. For example, to disambiguate between the "tears of joy" and "saddened" paths, ATLAST applies an evaluation metric between two competing paths of equal length that selects the oldest path. The Happy-State path was discovered first (arbitrarily, in this example), and thus remains as the interpretation of the input.

As their use of heuristic path-evaluators indicate, marker-passing systems generally permit themselves the luxury of using traditional symbolic buffers and programs to complement the spreading-activation process of the network. This allows them to build up complex symbolic representations of stories outside the network (as done by Norvig,[1989]) or hold rejected inference paths to allow reinterpretation if a path is rediscovered (as done by ATLAST when **Marriage** is followed by *Wilma was saddened by the proposal.*).

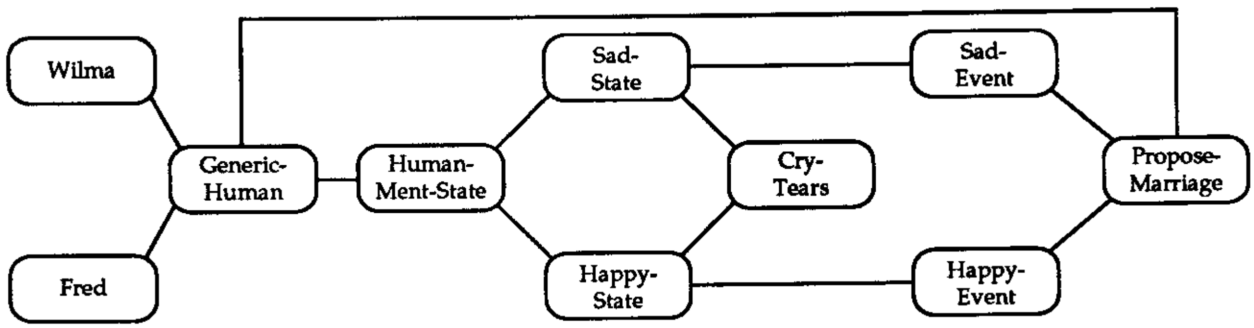

Figure 3. Marker-passing network from [Eiselt, 1987].

The best feature of marker-passing systems is that their parallel instantiation of inference paths makes them extremely efficient at generating different possible interpretations of the input. Unfortunately, the bottleneck for marker-passing systems is the separate path evaluation mechanisms used to select between generated interpretations (the heart of the disambiguation problem). The main problem is the extremely large number of *spurious* (i.e. non-important or logically-impossible) paths that the marker-passing process generates which the path evaluators must separately weed out. For even very small networks, these spurious paths often represent over 90 percent of the paths generated [Charniak, 1986]. More importantly, as the size of the networks increase to represent more world knowledge, there is a corresponding explosion in the number of paths generated. Because these paths must be evaluated serially by a path evaluator, it negates marker-passing systems' main efficiency advantage.

1.2. Abilities and Limitations of Each Connectionist Level

As can be seen in the above overview of how each connectionist level has been used for semantic language understanding, each level has a set of abilities and limitations that partially overlaps with the others. This is also true, of course, for their use in areas other than language understanding. A summary is presented in Table 1, which shows many of the processes necessary for cognitive modelling and how well each level of connectionist model performs them[4].

(1) *Self-Organization and Learning* — Distributed networks usually use learning algorithms such as back-propagation [Rumelhart *et al.*, 1986] to train their weights to map inputs to desired outputs. Similarity-based generalization and categorization are natural side-effects of the learning process. No such learning algorithms exist to organize localist networks, which must have their knowledge hand-coded, though adding to an existing network by recruitment of new units has been demonstrated [Diederich, 1990]. The knowledge in marker-passing networks and symbolic models must generally be hand-coded also, though many symbolic models learn through one-shot explanation-based learning (e.g. [Pazzani, Dyer, & Flowers, 1987]) or case-based reasoning (e.g. [Schank & Leake, 1989]).

(2) *Robustness to Noise and Damage* — Since knowledge in distributed networks is distributed over a large set of units, destruction or introduction of noise to a random subset of units, weights, or inputs has relatively little effect on input/output behavior (the models degrade gracefully). In contrast to this, destruction of

[4]A more detailed comparison between subsymbolic distributed network models and traditional symbolic models can be found in [Dyer, 1990].

Capability	Distributed Networks	Localist Networks	Marker-Passing Networks	Symbolic Models
Self-Organization and Learning	+	(−)	(−)	(+)
Robustness to Noise and Damage	+	−	−	−
Memory Blending and Interference	+	(−)	(−)	(−)
Associative Retrieval	+	+	−	−
Simple Processing Elements	+	+	−	−
Smoothly-Varying Commitment	+	+	−	−
Priming and Decay Effects	−	+	−	−
Complex Conceptual Relationships	(−)	+	+	+
Variable Bindings	(−)	(+)	+	+
Dynamic Constituent, Recursive Structure	−	(−)	+	+
Dynamic Inferencing	(−)	(+)	+	+
Knowledge-Level Parallelism	(−)	+	+	−
Meta-Reasoning	−	−	+	+

Table 1. Relative processing abilities of distributed connectionist networks, localist connectionist networks, marker-passing networks, and traditional symbolic models. + indicates something the class of models does relatively well. − indicates something the class of models does only with great difficulty, if at all. (+) and (−) indicate an ability demonstrated only recently or in a subset of the models.

random units or symbols in localist networks, marker-passing networks, and symbolic models causes permanent and localized damage to a specific piece of knowledge.

(3) *Memory Blending and Interference* — Human memory is far from perfect, with similar memories often blending or interfering with each other. Such blending occurs naturally in distributed networks, since similar memories share similar weight interconnection values and output activation patterns. On the other hand, although some symbolic models are able to partially explain confusions by assuming that different knowledge structures point to shared substructures, it is generally difficult to explain memory blending in localist networks, marker-passing networks, or symbolic models, since individual units or symbols perfectly represent a given memory.

(4) *Associative Retrieval* — Retrieving information in distributed networks and localist networks is generally done by clamping one or more of the network's inputs and allowing the network to settle into a state satisfying the largest number of constraints. Networks are often able to complete the pattern even when incomplete or noisy patterns are given as input, since the overall activation constraints from the partial pattern will likely be closest to those of the corresponding complete pattern. Retrieval in marker-passing networks is comparatively more brittle, because connections between units are generally all-or-none. Similarly, symbolic models generally retrieve only those items that match a set of explicit indices, so that any missing indices from partial input may rule-out retrieval. In general, it is more difficult for the binary nature of marker-passing networks and symbolic models to model the influence of contextual priming and varying influence of experience on retrieval.

(5) *Simple Processing Elements* — Distributed networks and localist networks are made up entirely of relatively simple numeric processing units whose activations based on their previous activations and the activations of their neighboring units [Feldman & Ballard, 1982]. Marker-passing networks, on the other hand, use more complex units that can hold the symbolic backpointers and structured information of markers, and often use labelled connections that perform different actions depending on the type or contents of those markers. Marker-passing networks, however, are still simpler than most traditional symbolic models, which generally use specialized rules and procedures to operate on and between knowledge structures.

(6) *Smoothly-Varying Commitment* — The graded activation levels and weights in distributed networks and localist networks allow them to have smoothly-varying levels of commitment to individual solutions, which can easily change given new biasing input. Marker-passing networks and symbolic models, on the other hand, generally use binary connections and rules so that a solution path can only be either active or inactive. Disambiguation and reinterpretation are thus more natural in distributed networks and localist networks, since each interpretation can have graded levels of activation, as opposed to marker-passing networks and symbolic models, in which disambiguation must be performed by separate (possibly conflicting) disambiguation heuristics.

(7) *Priming and Decay Effects* — Because activation on units in localist networks generally represents the amount of evidence available for concepts in a given context, their spreading-activation networks have been able to model many human priming effects, such as how people respond more quickly when presented with inputs similar to what they've just seen. Such priming effects are impossible in standard feed-forward distributed networks, because they have no record of their immediately-previous states. Since recurrent distributed networks do provide some record of their previous states, they do exhibit a form of priming, in that previous inputs influence future interpretations, but have no obvious way to model effects of priming on human reaction-times. Marker-passing networks and symbolic models are very awkward at modelling priming effects because of their general difficulties with smoothly-varying commitments.

(8) *Complex Conceptual Relationships* — Complex relationships between concepts (such as planning relationships) can readily be represented in localist networks and marker-passing networks by structured connections between units in the network. Such relationship rules are similarly direct to represent with pointers and rules in symbolic models. This is not the case in distributed networks, which have until recently had difficulty representing complex structured relationships, one of their primary limitations [Fodor & Pylyshyn, 1988]. These criticisms have been partially answered by recent rule-implementing distributed network models (cf. [Touretzky & Hinton, 1988]), and distributed models using recurrent networks and reduced descriptions (cf. [Pollack, in press]), but are still a problem.

(9) *Variable Bindings* — Marker-passing networks and symbolic models use their built-in abilities to allow variables to be dynamically bound to symbols for any type of structure, and new symbols (or markers) can be created during program execution. Localist networks, having only numeric activation levels, have only recently become able to hold variable bindings (cf. [Lange & Dyer, 1989], [Ajjanagadde & Shastri, 1989]), and are still limited with respect to symbolic models. Variable bindings have been even more problematic in distributed networks, but have also been shown to a limited degree in rule-implementing distributed networks. More traditional pattern-transformation distributed networks can also be trained to act as if they have variable bindings, but have the problem that bindings are often overridden by crosstalk from bindings that occurred often in the training set [St. John, 1990].

(10) *Dynamic Constituent, Recursive Structure* — Marker-passing networks and symbolic models can form an unlimited number of bindings without crosstalk, so can easily represent constituency and recursive structures, such as *John told Bill that Fred told Mary that...* Because recent localist network binding techniques

are limited in their binding capacity, it is difficult for them to represent such dynamic recursive structures, though Barnden [1990] and Holldobler [1990] have shown that it can be done, at least in untraditional localist networks. Recurrent distributed networks can be trained to learn *static* recursive structures (e.g. [Pollack, in press]), but have not yet been able to represent dynamic recursive structures that they have not been trained on.

(11) *Dynamic Inferencing* — Symbolic models have the ability to perform dynamic inferencing from an initial set of bindings by applying their rules to infer novel intermediate states having new bindings. Further inferences can then follow repeatedly from the new intermediate states until the desired state is reached. This ability is crucial when a system cannot reach the desired state in a single step [Touretzky, 1990]. Because marker-passing networks and recent localist networks can hold variable bindings and propagate them in turn for inferencing, they can also perform dynamic inferencing, though their inference rules are generally limited in complexity relative to symbolic models. Traditional pattern-transformation distributed networks cannot perform dynamic inferencing, since they transform the input (or set of inputs) to the output in a single step. Rule-implementing distributed networks, on the other hand, are able to perform a limited amount of dynamic inferencing.

(12) *Knowledge-Level Parallelism* — Marker-passing networks and localist networks are able to explore multiple solutions in parallel because alternative interpretations are represented by markers or activation patterns across different local areas of the network. This is crucial because with dynamic inferencing there are often a very large number of alternative solution paths, especially in language understanding and planning. In contrast, although distributed networks update their units in parallel, they are serial at the knowledge level because they represent all dynamic knowledge in a single set of units on the output or in a hidden layer, and so cannot make dynamic inferences from more than one potential solution at a time. Pattern transformation distributed networks can sometimes hold ambiguous solutions in a single blended activation pattern on their bank of hidden units, but it remains to be seen how far such blended activation patterns can be extended to simultaneously hold and make dynamic inferences from multiple, never-before encountered solution paths.

(13) *Meta-Reasoning* — Symbolic models can reason and operate on knowledge from many different substructures of their program, so long as that knowledge is represented by globally-interpretable symbolic structures. Marker-passing networks can do this also, since they often employ a separate high-level symbolic program to interpret and work with the results from the network's marker-passing process. Pure distributed networks and localist networks, however, attempt to complete their tasks entirely within the network, and so cannot perform meta-reasoning by resorting to symbolic code[5]. In addition, the knowledge encoded in the weights of the network (especially in distributed networks) is generally meant to perform the network's given task, and is therefore not as readily interpretable by external mechanisms. It is theoretically possible for distributed or localist network models to perform meta-reasoning within the network or with other subnetworks, but this is an area of connectionist research that has remained relatively unexplored.

[5]For practical purposes, results of the network are almost always analyzed by symbolic code or the human modeler. However, this analysis is rarely considered an integral part of the model.

2. THE CASE FOR HYBRID MODELS

All other things being equal, it is always desirable to build models out of as simple and homogeneous building-blocks as possible. This is one of the attractions of connectionist models in general, and distributed and localist connectionist models in particular, since their entire knowledge and processing mechanisms are built up of simple numeric processing elements and their local connections. It is also a reason against building hybrid models, since by definition hybrid models are made up of heterogeneous building-blocks from different connectionist levels that may or may not integrate naturally.

However, as the previous section illustrated, it is often not possible to build a successful model of a given cognitive task with elements from a single connectionist level given their current limitations. One is then confronted with a choice: abandon or scale back the task, attempt to extend the abilities of the connectionist level to handle the task, or utilize elements from another connectionist level to handle the task in a hybrid model. Obviously abandoning or scaling back the task is not always a desirable solution. Extending the abilities of the connectionist level to handle the task is perhaps ideally the best solution, and certainly a valuable long-term goal, but often requires theoretical breakthroughs that are not possible in a reasonable amount of time. Thus, if one is interested in building a model of human performance of a given task, then often the only possible approach is to build a hybrid model that combines elements and capabilities from multiple connectionist levels.

A number of researchers have recently argued that it is often desirable to build hybrid connectionist models (cf. [Dyer, 1990], [Hendler, 1989a], [Holyoak, in press], [Rose, 1990]). One argument is from an engineering perspective — if one's goal is to build a certain application or model without regards to the solutions' simplicity or elegance, then building a hybrid system is often the simplest (if not only) solution. This is true for real-world applications that require both low-level pattern-matching abilities and high-level symbolic abilities, such as automated manufacturing or testing applications which need both low-level visual perception and expert reasoning [Hendler, 1989a]. This is also true for researchers interested in modelling human performance, where matching psychological data is often more important than having a homogeneous model [Holyoak, in press].

Another reason for developing hybrid connectionist models is that they often turn out to be the most appropriate or useful level of description for complex systems. As Dinsmore [this volume] points out, high-level symbolic models are often the best description of processes that may actually happen on a lower (i.e. connectionist) level. The high-level abstractions of symbolic models allow predictions of cognitive behavior that would otherwise be too complex to understand. On the other hand, some processes do not lend themselves to higher-level abstractions, and are best described at a lower connectionist level. For large cognitive models that combine multiple such types of processing, the best model for descriptive and predictive processes is therefore often a hybrid in which each component is described and processed at its most useful level of description.

Even when using a single connectionist level is a major consideration, and a theoretical breakthrough seems possible to extend its abilities to handle a certain task, there are often good reasons to build a hybrid model first. If it appears necessary to have a certain ability, and that ability is present in another kind of network, then building a hybrid model of the two networks can serve as a useful prototype to validate the approach. If the hybrid model solves the problem or comes close to solving it, then it verifies that finding a way to embed the missing ability within the original network level will indeed be a fruitful solution. However, if the hybrid model having the ability in question is not able to solve the problem, then it serves as strong evidence that either a different approach is needed or that there are more facets to the problem than originally expected, and that the desired ability is not enough. Depending on the answer to this question, the hybrid model can therefore either save substantial effort trying to give the original network an unnecessary ability, or can shed light on other abilities that are needed to model the task.

2.1. Previous Hybrid Connectionist Models

Most connectionist research in cognitive modelling has involved building models out of a single connectionist level. Only recently have researchers begun to explore hybrid connectionist models as the advantages and disadvantages of each level have begun to become more clear.

Most work on hybrid connectionist models has been on networks that integrate marker-passing and localist connectionist techniques. A good example of the development of such models is the transformation of Hendler's [1988] marker-passing planning model to a model that uses both marker-passing and localist connectionist techniques (Hendler [1989b]). In Hendler's original marker-passing planning system, a symbolic problem-solving program would assert known facts and desired goals by placing markers into its semantic network memory. The marker-passing system then propagated those markers in parallel throughout the network, with intersections between markers reported back to the symbolic problem-solving program. The program would then evaluate the paths of concepts meeting at those intersections with a set of path-evaluating heuristics to determine whether they proposed a solution to an existing problem, caused a conflict, or otherwise provided useful information to the planner.

One of the main problems for Hendler's original marker-passing system was the rigidity of the underlying symbolic representation scheme that it (and all marker-passing systems) depend on. As in all marker-passing systems, there must be a link between two concepts for the connection between them to be found and used — for example, a link classifying a knife as a weapon. However, in many cases a connection between two concepts needs to be found, but would not normally exist as an explicit connection — for example, an ornate Egyptian letter opener would not normally be classified as a weapon, but could be considered one in certain contexts (since it is pointed, metallic, and sharp, like a knife). To more generally handle these kinds of cases, it is necessary to break the representation of concepts into the individual semantic features that describe them, something that causes problems in pure marker-passing systems because of the all-or-none nature of marker-passing paths. Hendler therefore expanded his networks to contain units representing needed semantic microfeatures and to include numeric *zorch* and threshold terms similar to the activation of localist connectionist networks (Hendler [1989b]). In the newer system, markers still propagate to find path intersections, but also hold numeric zorch terms representing their strength. Most importantly, these zorch amounts add up as activation on the individual units they reach, so that a unit that shares many features with another marked unit will receive a lot of activation and become part of the marker paths (e.g. a knife from a letter opener, since they are connected between the feature units for pointed, metallic, and sharp), but units that share only minimal features will not receive enough activation to support further marker propagation (e.g. a spoon from a letter opener, since it only shares the metallic feature). By including the analog evidence combination abilities of localist networks in his marker-passing system, Hendler was therefore able to solve a number of the representational problems of marker-passing systems and to approach a problem that neither level of modelling could perform well separately.

Kitano, Tomabechi, & Levin [1989] describe a model for natural language parsing and disambiguation that integrates marker-passing and localist connectionist techniques for much the same reason. In their system, marker-passing with three different kinds of markers is used to generate inferences and different hypotheses of a text's interpretation. One of the types of markers in their system serves as an "activation marker" that holds a numeric "cost" of the interpretation that it is a part of. The activation markers' costs increase whenever they mark previously inactive (unprimed) concepts or do not satisfy constraints imposed on their interpretation path. The activation markers over the path of units whose interpretation best matches the context of the story therefore tend to have the lowest costs, and are selected to represent the winning interpretation. This numeric summation of costs allows their system to integrate priming information and constraints to perform disambiguation better than purely-symbolic marker-passing systems. Like Hendler's [1989b] hybrid planning system, the integration of localist connectionist techniques with marker-passing into a hybrid model therefore enables processing that would be difficult in a purely marker-passing system.

Another hybrid marker-passing and localist connectionist network approach is to build localist networks that control their spread of activation using some of the symbolic techniques of marker-passing systems. Rose [1990] uses this approach in SCALIR, a model that performs conceptual retrieval of legal documents from a large semantic network. SCALIR uses a localist network whose units and interconnections represent concepts and documents in the legal domain, but whose units hold hybrid vectors of activation rather than a single activation value. Like connections between units in marker-passing networks, SCALIR's connections have symbolic labels that have different effects on the spread of activation depending upon their type. Connections with certain labels allow all components of the activity vector to be propagated through to the next unit after being multiplied by its weight, while connections with other labels let only a single specified component of the vector through. These different symbolic labels on SCALIR's connections allow the spreading-activation search process to be controlled symbolically, as in marker-passing systems, while retaining the associative retrieval and learning abilities of localist connectionist networks.

There have been fewer hybrid models that integrate localist connectionist and distributed connectionist networks. Sumida & Dyer [1989] have proposed a potential solution to the problem of distributed connectionist networks' being serial at the knowledge-level by integrating subnetworks of distributed ensembles into a large network that globally resembles a localist semantic network. Because each general concept is represented by a distributed ensemble of units, rather than the single unit they would be represented with in localist networks, they can be trained to store actual long-term memory instances of those concepts using distributed learning techniques, giving them an advantage over pure localist networks. On the other hand, the fact that the network is structured globally like a localist network (with ensembles for related concepts being connected to each other) gives their networks the potential to retain knowledge-level parallelism, an advantage over pure distributed networks.

2.2 Hybrid Connectionist and Symbolic Models

The largest class of hybrid models are models that combine connectionist networks with traditional symbolic processing. Such hybrid symbolic/connectionist models allow exploration of cognitive abilities that could not otherwise be handled in purely connectionist or symbolic systems, while being guides to best courses of future research. Most such models are distributed connectionist models that utilize symbolic abilities to handle portions of the task that the distributed networks cannot yet handle, such as the use of symbolic buffers in Kwasny & Faisal's [this volume] hybrid syntactic parser to store and manipulate the parse trees that their distributed networks are trained to operate on and build.

Hybrid symbolic/connectionist systems are especially valuable for functional approaches to model design, where a system is designed as a set of interconnected functional models that are first implemented symbolically but which are gradually replaced with connectionist modules. An example of this approach is DYNASTY [Lee, Flowers, & Dyer, 1990], a script-based story understander that uses multiple modules of recurrent distributed networks that access a symbolic dictionary. In their model, each distributed network is trained to serve as a module performing a separate processing subtask, such as parsing sequential input text into individual event representations, recognizing that a sequence of events fits into a particular script (e.g.

going to a restaurant), and paraphrasing the recognized script. Many of these distributed modules were initially implemented symbolically to allow testing of the overall model's concept, and were replaced one-by-one as time and opportunity presented it. The remaining symbolic component of DYNASTY is a symbolic hash table used as a "global dictionary" to store the representations of concepts and events it has learned and the symbols that represent them. Different distributed network modules access and store values in this symbolic global dictionary when needed.

An example of a hybrid symbolic/connectionist model that uses localist networks is Kintsch's [1988] construction-integration model of the psychological time course of language comprehension. Kintsch's model uses a symbolic production system to build symbolic representations of the alternative interpretations of a text and to construct a localist network in which the different interpretations compete. The spreading-activation process of the constructed localist network then serves to integrate the constraints from context (in the form of the excitatory and inhibitory connections constructed by the production system) to disambiguate and choose the correct interpretation. The use of the symbolic production system allows Kintsch's model to perform the rule-firing and inferencing that is difficult for purely localist models, while the constraint satisfaction of the constructed localist network allows modelling of the time-course of disambiguation that is difficult for purely symbolic models.

Finally, hybrid connectionist/symbolic models occasionally make use of multiple levels of connectionist processing in addition to their symbolic components. An example of this is Wermter & Lehnert's [1989] hybrid localist, distributed, and symbolic model for interpreting noun phrases such as *Note on the cause of ionization in the F-region*. Their model uses a symbolic syntactic parser to construct a localist network that represents the different possible combinations of nouns and prepositions for the input noun phrase. The localist network thereby constructed integrates the semantic and syntactic constraints to disambiguate the noun phrase and compute a preferred structural interpretation. The distributed connectionist networks are trained on the relative plausibility of semantic relationships between nouns, and are used initialize the activations of the localist network that does the actual disambiguation. By combining distributed networks, localist networks, and symbolic processing, their hybrid model allows for a combination of learning, integration of competing constraints, and symbolic extraction of concepts difficult for models that use only one of the three types of processing.

3. GOING FROM HYBRIDS TO A SINGLE LEVEL

If hybrid connectionist models are to be of value to researchers ultimately interested in building models on a single connectionist level, then it must eventually be possible to at least roughly re-create the hybrid capabilities in that level. Even a successful hybrid model can turn out to be little more than an engineering exercise for a researcher interested in pure connectionist models if it turns out to be impossible to implement the hybrid in the desired level. For example, a hybrid symbolic/connectionist model for playing chess that uses symbolic routines to perform brute-force search of the game tree and a distributed network as a pattern-matcher to evaluate the board positions might be a perfectly reasonable engineering approach to building a competitive chess-playing program. However, it would probably fail as a significant stepping-stone to a purely-connectionist model of human chess playing, since it seems unlikely that any connectionist mechanism will be able to perform the millions of individual search and evaluation steps needed to implement the hybrid's brute-force search with human response times.

On the other hand, if the elements and mechanisms borrowed from a foreign connectionist level in a hybrid model have things in common with the desired connectionist level, then it is more likely that the model will eventually be implementable in that level. If it is indeed implementable, then the hybrid model will have fulfilled its mission of confirming that those mechanisms will be useful to the task. When a mechanism is actually developed to allow the hybrid model to be implemented in a single level, it will likely have advantages over the hybrid model (besides its homogeneity), since the new mechanism within the level will likely integrate more smoothly with the rest of the level than the sometimes artificial interface between elements of different levels in hybrid models (such as between numeric activation and symbolic markers).

For example, suppose a researcher has built a hybrid localist connectionist and marker-passing model to test how well a localist connectionist network would work for language understanding if it had a marker-passing network's ability to hold variable bindings and perform inferencing. If the hybrid model works well, then it would confirm that it would be valuable to have a marker-passing network's variable binding abilities in localist connectionist networks. However, to complete the circle and make the whole effort worthwhile for a researcher mainly interested in localist connectionist networks, it must turn out to indeed be possible to implement the marker-passer's variable binding abilities within a purely localist connectionist network.

3.1. ROBIN: A Localist Connectionist Network With Hybrid Abilities

As described in Section 1.1, semantic language understanding is such a large and difficult task that no level of connectionist or symbolic processing can perform many aspects of it particularly well. Localist connectionist networks seem best-suited to handling the ambiguity rife in language, but have had no way to represent the variable bindings and perform the inferencing necessary for comprehension. Marker-passing networks are well-suited for performing inferencing, but are awkward when it comes to resolving ambiguities.

An obvious solution to this dilemma is to build a hybrid marker-passing and localist connectionist network that propagates markers to generate alternative inference paths and uses the constraint-satisfaction of the localist network's activation to disambiguate between those inference paths. Such a hybrid model is relatively straightforward to build because of the similarity of the unit structure in marker-passing and localist networks. The two types of propagation can in fact proceed in parallel across a single set of hybrid units that can hold both activation and markers. Lange, Hodges, Fuenmayor, & Belyeav. [1989] briefly describe a simple such hybrid model that performs both inferencing and disambiguation within the network. Kitano *et al.* [1989] also describe a hybrid marker-passing and localist connectionist model that performs parsing and disambiguation.

Given that a hybrid marker-passing and localist connectionist model has many potential advantages for semantic language understanding, it would be desirable to build a purely-localist connectionist model that has the variable binding and inferencing abilities of marker-passing networks while retaining its disambiguation abilities.

ROBIN (ROle Binding and Inferencing Network) [Lange & Dyer, 1989] is a purely-localist, non-hybrid connectionist model that has many of the variable binding inferencing abilities of marker-passing networks. Because ROBIN also retains the disambiguation abilities of normal localist networks, it is able to perform high-level inferencing that requires lexical and pragmatic disambiguation. As an example of the kinds of input ROBIN is able to understand, consider the phrase:

P1: *John put the pot inside the dishwasher*

I1:	If the police see John's marijuana, then they will know that he possesses an illegal object (since marijuana is an illegal substance).
I2:	If the police know that John is in possession of an illegal object, then they will arrest him, since possessing an illegal object is a crime.
I3:	John does not want to get arrested.
I4:	John has the goal of stopping the police from seeing his marijuana.
I5:	The police coming results in them being in the proximity of John and his marijuana.
I6:	The police being in the proximity of John's marijuana enables them to see it.
I7:	John's putting the marijuana inside the dishwasher results in the marijuana being inside the dishwasher.
I8:	The marijuana is inside an opaque object (the dishwasher).
I9:	Since the marijuana is inside an opaque object, the police cannot see it, thus satisfying John's goal.

Table 2. Inferences ROBIN makes to understand the sentence *John put the pot inside the dishwasher because the police were coming.* **(Hiding Pot)**

To understand **P1**, ROBIN disambiguates the word *pot* to mean a Cooking-Pot, and infers that the most likely reason for John putting it inside the dishwasher was to get it clean. However, later context often shows the original inferences to be wrong, forcing reinterpretation of the input. This is the case if **P1** is followed by:

P2: *because the police were coming.*

Suddenly, the best interpretation for *pot* in P1 changes to Marijuana, and John's Transfer-Inside action seems to be a plan for hiding the Marijuana from the police to avoid his arrest. This reinterpretation can only be made after generating a chain of inferences to find the causal relationship between the two phrases (collectively referred to as **Hiding Pot**). Table 2 shows the inferences ROBIN makes to dynamically create this interpretation.

3.1.1. Structure of Robin

ROBIN's networks consist entirely of connectionist units [Feldman & Ballard, 1982] that perform simple numeric computations on their inputs: summation, summation with thresholding and decay, or maximization. Connections between units are weighted, and either excitatory or inhibitory. These networks encode semantic networks of frames representing world knowledge. Each frame has one or more roles, with each role having expectations and selectional restrictions on its fillers. Every frame is related to one or more other frames, with pathways between corresponding roles (representing general knowledge rules) for inferencing. This section gives a short overview of ROBIN and how it performs inferencing, but [Lange & Dyer, 1989] provides a detailed description.

As in most localist connectionist models, there is a single unit in the network for each frame or role concept in the knowledge base, with relations between concepts being represented by weighted connections between the units. Activation on a conceptual unit is *evidential*, corresponding to the amount of evidence available for the concept (either a frame or role) and the likelihood that it is selected in the current context.

As described before, representing the amount of evidence available for a concept, however, is not sufficient for complex inferencing tasks. A solution to the variable binding problem requires that some means exist for *identifying* a concept that is being dynamically bound to a role, as marker-passing networks do with the

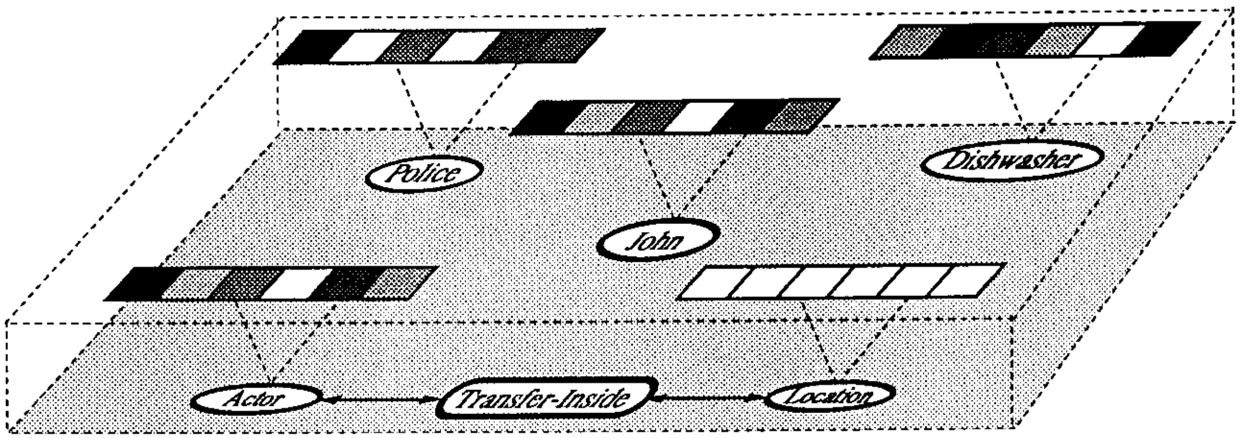

Figure 4. Several concepts (ovals on lower plan) and their uniquely-identifying signature patterns, along with the Actor and Location roles of the Transfer-Inside frame. Here each signature is a bank of six units, with increasing levels of activation represented by increasing darkness of shading (ranging from white = 0 to black = 1). The Actor role has a *virtual binding* to John because its binding units hold the same activation pattern as John's signature. The Location role shown here is currently unbound (binding banks have no activation).

symbolic backpointers on their markers. Furthermore, the network's structure must allow these role-bindings to propagate across unit pathways that encode the knowledge base's rules (as do markers), thus dynamically instantiating inference paths representing the input.

3.1.2. Variable Binding With Signatures *In Localist Connectionist Networks*

Representing variables and role-bindings is handled in ROBIN by network structure holding *signatures* — activation patterns which uniquely identify the concept bound to a role (introduced in [Lange & Dyer, 1988]). Every concept in the network has a set of *signature* units that output its signature, a constant activation pattern different from all other signatures. A dynamic binding exists when a role or variable's *binding units* have an activation pattern matching the activation pattern of the bound concept's signature. For example, in Figure 4, the *virtual binding* of the Actor role of action Transfer-Inside (representing somebody putting an object inside another, as in P1) to John is represented by the fact that its binding units have the same activation pattern as John's signature. The same binding units could, at another time, hold a different virtual binding, simply by having the activation pattern of another concept's signature. The complete Transfer-Inside frame is represented in the network by the group of units that include the conceptual unit Transfer-Inside, a conceptual unit for each of its roles (the Object role not shown), and the binding units for each of its roles.

In general, signatures can be uniquely-identifying activation patterns of any size. Ideally, signatures are distributed activation patterns (e.g. made up of semantic microfeatures) that are themselves partial representations of the concept they stand for. This allows the signatures themselves to be used as inputs for distributed learning mechanisms after they have been propagated for inferencing. For simplicity, however, ROBIN's simulations are usually run with the signature patterns simply being arbitrarily-generated scalar values that uniquely identify their concept.

3.1.3. Propagation of Signatures for Inferencing

The most important feature of signatures is that they can be propagated without change across long paths of binding units to dynamically instantiate candidate inference paths. Connections between binding units of frames' roles encode rules such as:

```
R1:  [Actor X Transfer-Inside Object Y Location Z]
        == results-in ==> [Object Y Inside-Of Location Z]
```

(When an object is transferred inside of a location, then it is inside of that location)

Figures 5a and 5b illustrate how the network's structure automatically propagates signatures to fire rules such as R1. For simplicity, the signatures in the figure are uniquely-identifying scalar values. Evidential activation for disambiguation is spread through the paths between conceptual units on the bottom plane (i.e. Transfer-Inside and its Object role), while signature activation for dynamic role-bindings is spread across the parallel paths of corresponding binding units (solid black circles) on the top plane. Units and connections for the Actor, Planner, and Location roles are not shown. As shown in the figure, there are actually multiple binding units per role to allow simultaneous propagation of ambiguous bindings. In general, this requires that there be as many binding units per role as there are possible meanings of the most ambiguous word in the network.

Initially there is no activation on any of the conceptual or binding units in the network. When input for *John put the pot inside the dishwasher* (P1) is presented, the lexical concept units for each of the words in the phrase are clamped to a high level of evidential activation, directly providing activation for concepts John, Transfer-Inside, Cooking-Pot, Marijuana, and Dishwasher. To represent the role-bindings given by phrase P1, the binding units of each of Transfer-Inside's roles are clamped to the signatures of the concepts bound to them[6]. For example, the binding units of Transfer-Inside's Object are clamped to the activations (6.8 and 9.2) of the signatures for objects Marijuana and Cooking-Pot, representing the candidate bindings from the word *pot* (Figure 5a)[7].

The activation of the network's conceptual units is equal to the weighted sum of their inputs plus their previous activation times a decay rate, similar to the activation function of previous localist networks. The activation of the binding units, however, is equal to the maximum of their unit-weighted inputs, allowing signatures to be propagated without alteration. Binding units calculate their activation as the maximum of their inputs because this preserves their signature input value even when the signature can be inferred from more than one direction. The actual relative signature activation values do not matter, since gated connections (not shown) ensure that two different signatures do not reach the same binding node [Lange & Dyer, 1989].

---

[6] ROBIN does not currently address the problem of deciding upon the original syntactic bindings, i.e. that *"pot"* is bound to the Object role of phrase P1. Rather, ROBIN's networks are given these initial bindings and use them for high-level inferencing.

[7] An alternative input, such as *"George put the cake inside the oven"*, would be done simply by clamping the signatures of its bindings (i.e. George, Cake, and Oven) instead. A completely different set of inferences would then ensue.

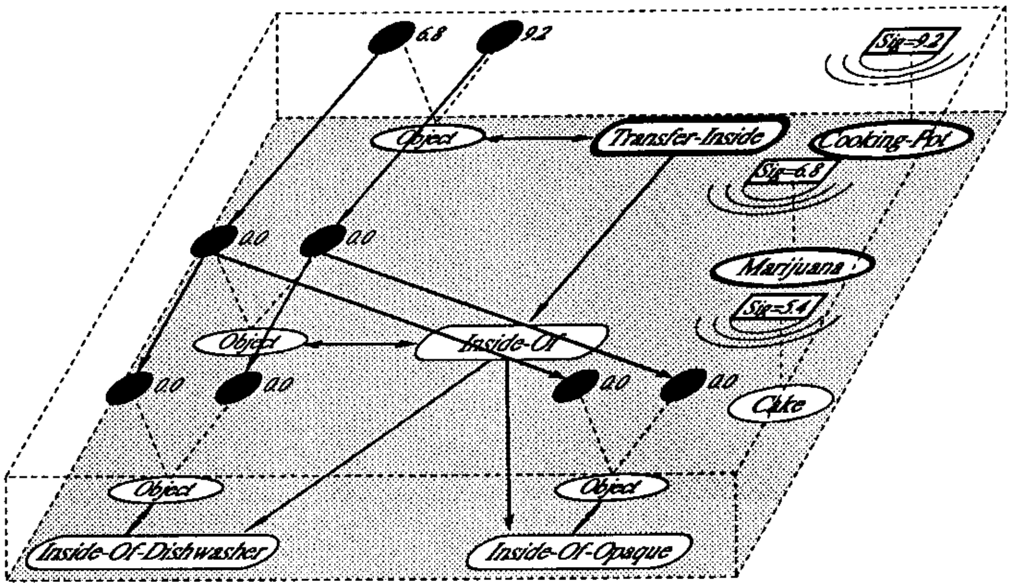

Figure 5a. Initial activation for **P1.**

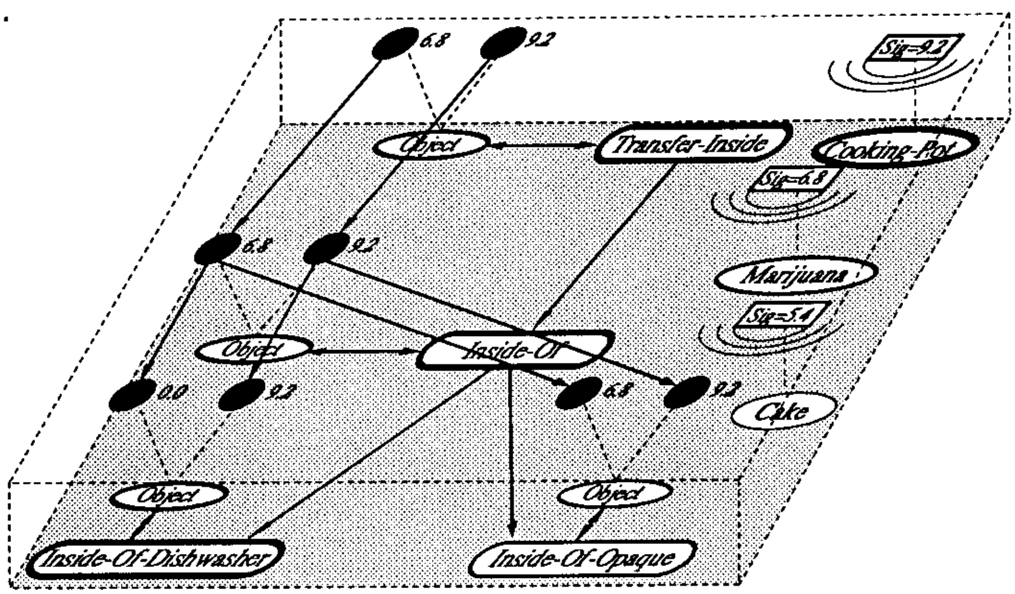

Figure 5b. Activation after quiescence has been reached in processing for **P1.**

Figure 5. Simplified ROBIN network segment at two different cycles during processing of **P1** (*John put the pot inside the dishwasher*). Each figure shows the parallel paths over which evidential activation (bottom plane) and signature activation (top plane) are spread for inferencing. Signature nodes (outlined rectangles) and binding nodes (solid black circles) are in the top plane. Thickness of conceptual node boundaries (ovals) represents their levels of evidential activation. (Node names do not affect the spread of activation in any way. They are simply used to initially set up the network's structure and to aid in analysis.)

As activation starts to spread after the initial clamped activation values in Figure 5a, Inside-Of receives evidential activation from Transfer-Inside, representing the strong evidence that something is now inside of something else. Concurrently, the signature activations on the binding units of Transfer-Inside's Object propagate to the corresponding binding units of Inside-Of's Object (Figure 5b), since each of the binding units calculates its activation as the maximum of its inputs. For example, Inside-Of's left Object binding unit has only one input connection, that from the corresponding left Object binding unit of Transfer-Inside. Since the connection has a unit weight and the left Object binding unit of Transfer-Inside has an activation of 6.8, Inside-Of's left Object binding unit also becomes 6.8 (Marijuana's signature), because 6.8 is its maximum (and in this case only) input. The potential binding of Cooking-Pot (signature 9.2) to Inside-Of's right Object binding unit propagates at the same time, as do the bindings of Inside-Of's Planner role to the signature of John and its Location role to the signature of Dishwasher.

The network has thus made the crucial inference of exactly which thing is inside of the other, by propagating signatures across binding paths encoding rule R1. Similarly, as time goes on, Inside-Of-Dishwasher (representing a kitchen utensil being inside of a dishwasher, a precondition for cleaning) and Inside-Of-Opaque (representing an object being inside of an opaque object, which blocks it from sight) receive evidential activation, with inferencing continuing by the propagation of signature activation to their corresponding binding units (Figure 5b)[8].

Inferencing continues by propagation of signature and evidential activation. Figure 6 shows an overview of the signature bindings in a portion of the network after input for the rest of **Hiding Pot** (*because the police were coming*) is presented and the network eventually settles. The network has made inferences I1-I9 of Table 2, with most being shown in the figure. For example, I8 (the inference that the Marijuana is inside of an opaque object) is represented by the instantiation of state Inside-Of-Opaque. The role-bindings of the frames shown were instantiated dynamically with signature activation.

3.1.4. Disambiguation and Reinterpretation

As can be seen in figures 5 and 6, propagation of signature activations dynamically instantiates candidate inference paths in parallel in much the same way that marker-passing systems do. If this were a marker-passing system, then an external symbolic path evaluator would have to be used to select between the dishwasher cleaning path and the longer hiding path connecting John's Transfer-Inside to the Police's Transfer-Self. The evaluation heuristics would also have to somehow recognize that at the end of processing Marijuana should be selected over the Cooking-Pot and Planting-Pot bindings throughout the network.

However, in ROBIN, such disambiguation is performed entirely within the network, without the need to resort to a separate path-evaluation program. Deciding between the competing inference paths instantiated by signature activation is the function of the evidential portion of ROBIN's networks (such as the concep-

[8]Inside-Of-Dishwasher and Inside-Of-Opaque are *concept refinements* (or specializations) of Inside-Of. Refinement frames here represent the *reason* for a particular action or state, and are useful because they allow more specific inferences to be made when role-bindings are known. For example, if the network has inferred that a dish is inside of a dishwasher (Inside-Of-Dishwasher), then it could infer that it is going to be cleaned. If the network has inferred that any object is inside of an opaque object (Inside-Of-Opaque), then the network could infer that what is important is that the object is blocked from sight. When more than one refinement of a frame can be inferred (as in **Hiding Pot**), one of them must be selected as the winning reason in the given context (e.g. is the object being cleaned or hidden?).

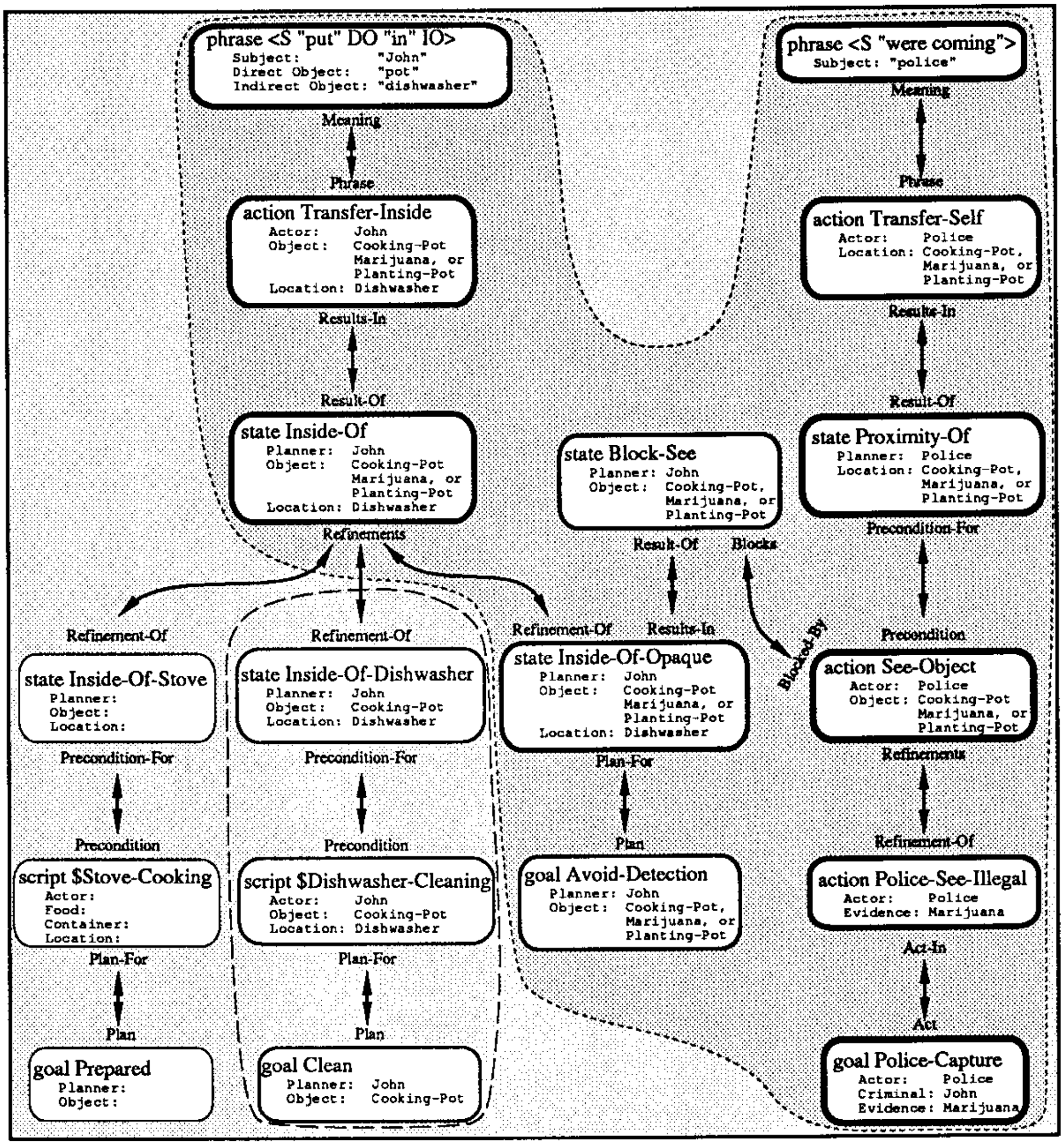

Figure 6. Overview of a small portion of a ROBIN semantic network (actually embedded in network structure such as in Figures 5a and 5b) showing inferences dynamically made after clamping of the inputs for phrases P1 and P2 of **Hiding Pot**. Thickness of frame boundaries shows the amount of *evidential* activation on the frames' conceptual nodes. Role fillers shown are the ones dynamically instantiated by propagation of *signature* activation over the role's binding nodes. Darkly shaded area indicates the most highly-activated path of frames representing the most probable plan/goal analysis of the input. Dashed area shows the discarded dishwasher-cleaning interpretation. Frames outside of both areas show a very small portion of the rest of the network. These frames received no evidential or signature activation from either phrase.

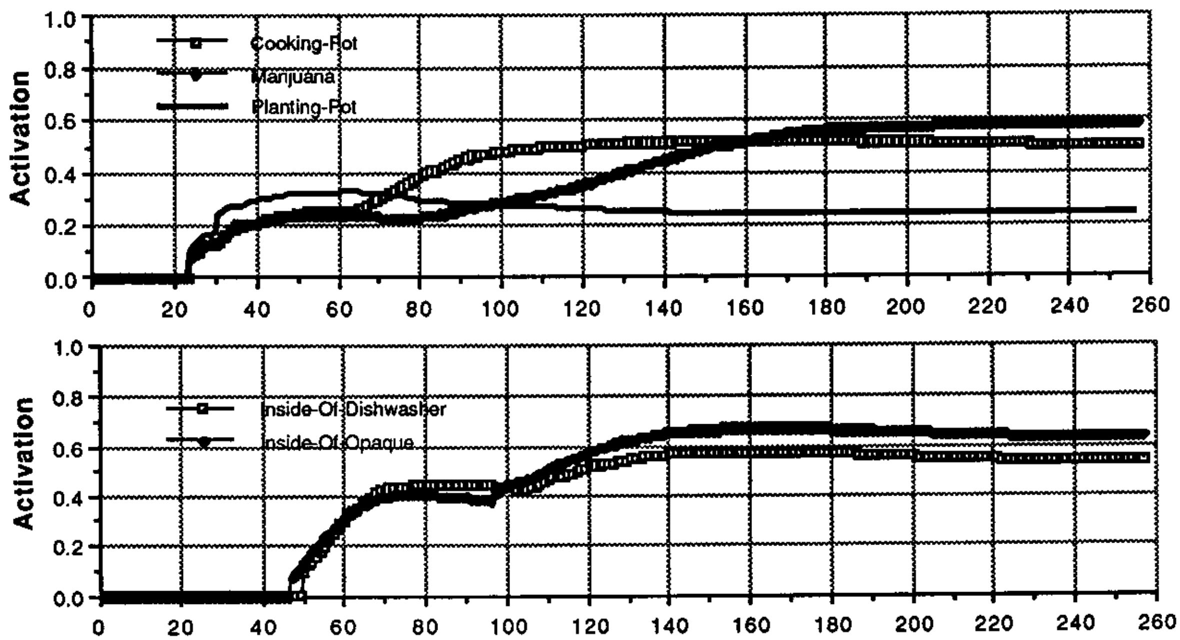

Figure 7. Time-course of evidential activations of meanings of word *pot* and competing refinements of Inside-Of after presentation of input for *John put the pot inside the dishwasher* (**P1**) at cycles 1 through 31 and presentation of input for *the police were coming* (**P2**) is presented at cycles 51 through 61.

tual units on the bottom layer of Figures 5a and 5b). The activations of the conceptual frame units are always approximately proportional to the amount of evidence available for them from their bindings and their related frames. The inference path selected as the interpretation in any given context is therefore simply *the most highly-activated path of frame units* and their bindings[9]. Similarly, when there are multiple possible bindings for each role, the binding chosen at any given time is the one whose concept has the highest level of evidential activation.

ROBIN has been implemented in the DESCARTES connectionist simulator [Lange *et al.*, 1989]. Figures 7a and 7b show the evidential activations of the ambiguous meanings of the word *pot* and the competing refinements of Inside-Of as activation spreads through the network. Initially there is more evidence for the interpretation that John was trying to clean a cooking pot. This is shown by the fact that Cooking-Pot becomes more highly-activated than Marijuana or Planting-Pot after Inside-Of-Dishwasher becomes activated (about cycle 60). However, after input for P2 is presented at cycles 51 through 61, the inferences about the police propagate through Transfer-Self, Proximity-Of, See-Object, and Block-See, until they reach Inside-Of-Opaque (about cycle 95), as in Figure 6. Reinforcement from this hiding and police capture path eventually causes Inside-Of-Opaque to become more highly-activated than Inside-Of-Dishwasher and Marijuana to become more highly-activated than Cooking-Pot (by cycle 160), so that the network's interpretation of the input changes to the hiding marijuana interpretation of the darkly-shaded area in Figure 6. Notice,

[9]The network's "decision" or "selection" is actually simply the interpretation that the human modeler gives to the levels of activation present in it, as in all connectionist models.

however, that evidential activation remains on the units of the alternative paths, allowing another possible reinterpretation if the next sentence is *they were coming over for dinner in half an hour.*

3.1.5. Elimination of Crosstalk: Interaction of Signature and Evidential Activation

So far, signature and evidential activation have been described as propagating in parallel but along separate paths of units and connections. However, as described in [Lange, in press], the problem of *crosstalk* makes it crucial for the two paths of activation interact so that the dynamic variable bindings in the network affect the spread of activation.

One way ROBIN controls crosstalk is by having units embedded within it that compute and enforce *selectional restrictions* on role-fillers to control the spread of activation when roles' binding constraints are violated. For example, the selectional restrictions on the Object role of Inside-Of-Dishwasher expect it to be filled only by objects that are cooking or eating utensils, and not objects like Marijuana. To enforce these selectional restrictions, each connection from one binding unit to another is actually a *multiplicative connection* (as in the sigma-pi units described in [Rumelhart *et al.*, 1986]) that is *gated* by another unit calculating whether the signature is a legal one. For example, the left gated link from the binding unit of Inside-Of's Object to Inside-Of-Dishwasher's Object in Figure 5b is closed, since the network recognizes that Marijuana (6.8) violates its selectional restrictions. Only the signature of Cooking-Pot (9.2) matches and is propagated to be considered as the Object of Inside-Of-Dishwasher.

In other cases, the role-filler's constraints on a frame are completely violated (e.g. Inside-Of-Stove and Inside-Of-Restaurant are impossible interpretations for P1). In these cases, the activations of the signature bindings interact with the activation on the evidential layer through gated connections that stop the violated frames from receiving activation, as can be seen in Figure 6. These selection restrictions (or *logical binding constraints*) dramatically reduce the number of spurious inference paths generated by the propagation of signatures and thus eliminate a large potential source of crosstalk. The network structure imposing selectional restrictions is not important for the purposes of this paper, but is described in [Lange & Dyer, 1989].

Another way in which the activation of the signature role-bindings and the activation of the evidential layer interact is by structure that assures that evidential activation is spread only between frames and their *actual* role-fillers. This is to solve localist connectionist networks' basic problem of not being able to distinguish between sentences such as *The astronomer saw the star* and *The star saw the astronomer* (section 1.1.3). Signatures partially solve this problem by allowing the network to differentially represent the bindings of the two different instances. However, if these bindings do not have an effect on the spread of evidential activation, then they might as well not be there in terms of disambiguating between the meanings of *star*. ROBIN solves this problem by gated connections that feed evidential activation back from a frame to those concepts, and only those concepts, that are bound to its roles with signature activation. In the case of *The star saw the astronomer*, only the signature of Movie-Star reaches the Actor role of See (since Celestial-Body violates its selectional restrictions). Movie-Star therefore receives evidential activation that Celestial-Body does not, so that it becomes more highly-activated and is chosen as the interpretation of *star*. This control of activation based on signature bindings is also done by structures of units and gated connections within the network; how crucial it is to the disambiguation process by controlling crosstalk is explained more thoroughly in [Lange, in press].

3.2. Comparison of ROBIN and Hybrid Models

Signatures allow ROBIN to hold and propagate variable bindings much like symbolic markers allow marker-passing networks to. Because they are simply activation patterns that spread across normal numeric connectionist units and connections, they allow purely-localist connectionist networks to have much of the functionality of hybrid localist and marker-passing networks. Equally important is that ROBIN retains the normal disambiguation abilities of localist networks, unlike other localist models that have demonstrated the ability to handle variable bindings (such as [Ajjanagadde & Shastri, 1989], [Barnden, 1990], and [Holldobler, 1990]). ROBIN thus completes the circle; hybrid networks such as those of Lange *et al.* [1989] and Kitano *et al.*, [1989] demonstrate that localist networks having marker-passing abilities are useful for language understanding, and ROBIN demonstrates that the abilities of these kinds of hybrid models can actually be embedded within a purely-localist network .

As is generally the case when a mechanism is developed to give a specific connectionist level the capabilities of hybrid model, the signature mechanism for variable bindings in localist networks has both advantages and disadvantages in comparison to that of marker-passing in hybrid networks. One weakness of signatures is that each binding unit can hold only one signature activation at a given time, while each marker-passing unit can hold as many markers as its symbolic stack can hold. This is why each role of ROBIN's frames has multiple binding units to hold ambiguous bindings (such as of the word *pot*). Another difference between signatures and markers is that signature activation patterns only represent the concept being bound, while markers can also hold complex symbolic information such as the type of the marker, the path it has followed, the time the marker arrived, and so on.

On the other hand, a purely-localist connectionist model such as ROBIN has the advantage that its building-block elements are all relatively simple, numeric connectionist elements. This is in contrast to hybrid localist and marker-passing networks, whose elements must not only support normal connectionist activation functions, but must also be capable of holding lists of symbolic markers and acting on the sometimes complex symbolic information on those markers. The more important advantage of ROBIN's purely-localist networks, however, lies in how naturally signature activation variable bindings interact with evidential activation as opposed to the variable bindings held in symbolic markers. For example, a signature matches a selectional restriction if its concept has been inferred to be an instance of the type of that restriction (e.g. if the Dishwasher in the phrase has been inferred to be an Instance of type Opaque-Object). This is calculated in the network by comparing the signature activation on the candidate binding unit to the signature activation on each of the binding units of the restriction type's Instance role (by units having opposite-signed weights from each and a low firing threshold). If any match, then the signature is of the right type, the restrictions are met, and the corresponding binding constraint unit becomes active. A multiplicative connection from this binding constraint unit to the connection on the evidential layer from one frame to the other is then all that is needed to gate the flow of evidential activation open and closed when necessary. The symbolic information held in the activations of the signature bindings thereby controls the disambiguating activations of the evidential layer by the normal spreading-activation process.

In a hybrid marker-passing/localist connectionist model, on the other hand, there is nothing akin to multiplicative connections to cleanly interface between the symbolic information on markers and the numeric information of evidential activation. Thus, to enforce selectional restrictions within the network, the weighted activation connections between frames would have to somehow query or be controlled by the symbolic marker-passing units (which would have to symbolically calculate whether the restrictions have been met). This is certainly possible in a hybrid model, but having such awkward communication between different processing paradigms makes the network more complex and violates the normal numerically-based activation functions of connectionist units [Feldman & Ballard, 1982].

A more fundamental potential advantage to using activation-based signatures in a localist network as opposed to using a hybrid with marker-passing comes into play if signatures are uniquely-identifying distributed patterns of activation (as shown in Figure 4) rather than the arbitrary scalar values shown propagated in Figures 5a and 5b. In this case, the distributed representations of similar concepts would have similar signature patterns and thereby carry a degree of semantic information that could be used locally in the network. Such a move towards a hybrid localist and distributed connectionist model would allow signature inferencing to automatically drive distributed learning of selectional restrictions and long-term instances (as proposed in [Lange & Dyer, 1989]), something not possible with the symbolic backpointers of markers. By finding a mechanism that allows localist networks to handle the variable-binding and inferencing of hybrid marker-passing and localist connectionist models, we have thereby moved closer to being able to naturally take advantage of the features of the third level — distributed connectionist networks.

4. HYBRID COGNITIVE MODELS

Most connectionist and artificial intelligence systems attempt to model one subtask of a single cognitive ability, such as of natural language understanding, planning, memory retrieval, or visual processing. Given the current relatively primitive understanding of how to model cognitive functions, this is generally the best way to explore an area in any depth. In people, however, processing of different cognitive functions are rarely completely separate, and the interaction of two or more cognitive functions can have dramatic effects on each other. What people see or read affects what they think, what plans they make, what they say, and what they remember. Because different types of cognitive functions tend to range in the amounts of symbolic and perceptual (subsymbolic) processing they require, models that attempt to explore two or more cognitive functions and how they interact with each other are especially likely to benefit from a hybrid connectionist modelling approach.

For example, natural language understanding and memory retrieval are two tightly-intertwined cognitive functions. When a person understands a text, he is using his natural language understanding abilities to build a conscious interpretation of the text's meaning. This interpretation will often trigger a reminding of a similar or analogous episode from memory, which may in turn be used to aid in his understanding of the text, bolster an argument, or simply sidetrack the person to think of something more interesting. The language understanding process therefore affects the memory (or analogical) retrieval process, which, in turn, affects language understanding by changing the context in which the next pieces of text or speech will be disambiguated and understood.

Although there have been a number of symbolic and connectionist models of language understanding and analogical retrieval, few have dealt with how the two processes are integrated and affect each other. This section gives an overview of SAARCS (Spreading-Activation Analog Retrieval by Constraint Satisfaction) [Lange, Melz, Wharton, & Holyoak, 1990], a hybrid localist connectionist and marker-passing network that performs both language understanding and analogical retrieval in order to model the effects of inferencing and disambiguation on the memory retrieval process. The fact that SAARCS is a hybrid model that combines elements from both localist and marker-passing networks allows it to explore aspects of this problem that would currently be difficult to explore in a pure model from either level, since it ranges from handling low-level priming and disambiguation (difficult to model in marker-passing networks) to large-scale comparisons of symbolic structure (difficult for localist networks).

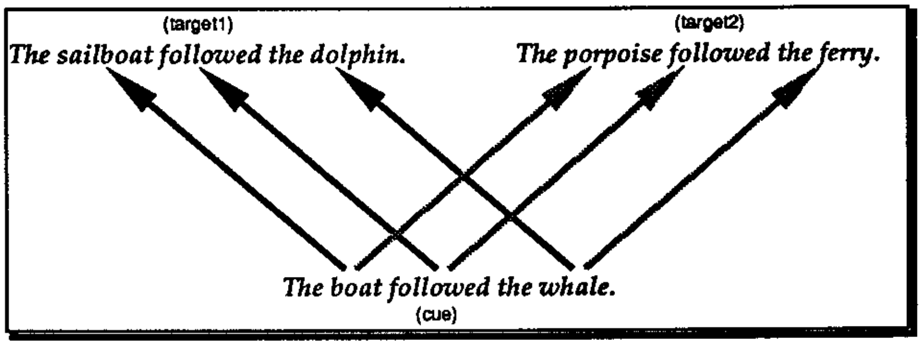

Figure 8. Cue that shares similar concepts with two targets, but maps consistently to one only (heavy arrows).

4.1. Analogical Retrieval

Human memory retrieval involves more than just matching text against items in memory. Comprehension processes, such as disambiguation and inferencing, will alter the effective retrieval cue. Thus a realistic model of episodic reminding must integrate the process by which the retrieval cue is understood with the process by which it is used to recall information from memory.

Considerable evidence indicates that a primary influence on reminding is the degree of direct semantic similarity between the cue and objects in memory [Holyoak & Koh, 1987] [Ross, 1989]. Though the evidence is not as compelling as for semantic similarity, some recent work has shown that structural consistency (i.e., analogy) also influences the retrieval process. *Structural consistency* requires that if two frames are placed in correspondence, then their roles and fillers should also correspond [Holyoak & Thagard, 1989]. Figure 8 illustrates a simple example of variation in structural consistency. Suppose a person has studied the sentences *The sailboat followed the dolphin* and *The porpoise followed the ferry*, and is then cued with *The boat followed the whale*. If the cue is viewed as being mapped to potential targets (or episodes) in memory, then the former target yields a consistent mapping in which similar objects fill the corresponding agent and object roles, whereas the latter target generates an inconsistent cross mapping in which similar objects play dissimilar roles. Ross [1989] found that cross mapping impaired retrieval of formulas to solve story problems when the analogs involved similar objects.

We believe, however, that the effect of analogy on reminding will be influenced by several other factors. First, cue/target semantic similarity is a necessary condition for structural consistency to affect reminding. If two situations are dissimilar, then the retrieval cue will likely fail to make contact with (activate) a stored representation of the individual concepts, in which case configural properties will be irrelevant. Second, there is considerable evidence that human memory is sensitive to retrieval interference effects (e.g. [Nickerson, 1984]). Because of retrieval competition, a stored potential analog that maps inconsistently to the retrieval cue may be less likely to be recalled if a rival analog with a consistent mapping to the cue is also stored in memory (see Figure 8).

Finally, the impact of structural consistency and retrieval competition influences and is influenced by the comprehension processes involved in lexical disambiguation. The reversal of case role fillers, which can alter the structural consistency of a mapping, can also alter preferred interpretations of individual lexical items. For example, the fish in *The surfer ate the fish* is small, dead, and cut up, whereas the fish in *The fish ate the surfer* is very large, alive, and whole. In such cases, a role reversal can effect the interpretation of lexical items, which in turn can alter the similarity of individual concepts in the cue to the concepts in a stored potential analog, as well as altering configural resemblance. The inferences needed for the comprehension process are also crucial to retrieval — without a minimal understanding of how the concepts and actions of a cue are related, it is unlikely that a reasoner will retrieve a proper analogy to a given cue. This is especially true if the potential memories are indexed in ways that can only be inferred indirectly from the cue.

4.2. SAARCS: A Hybrid Connectionist Model of Understanding and Retrieval

SAARCS is a hybrid localist connectionist and marker-passing model that integrates language comprehension and analogical retrieval. Given the syntactic representation of an input sentence as a cue, SAARCS' network first disambiguates and infers an interpretation of the cue, and then retrieves and returns the sentence or episode from long-term memory that is analogically closest to that interpretation. The system combines the ROBIN localist connectionist model for disambiguation and inferencing [Lange & Dyer, 1989] with aspects of ARCS, a hybrid symbolic/localist connectionist model of analog retrieval [Thagard, Holyoak, Nelson, & Gochfeld, in press]. Because we are interested in modelling the processes of disambiguation and inferencing and their effects on analogical retrieval, SAARCS combines marker-passing with a localist spreading-activation network in a single integrated model.

SAARCS consists of a localist connectionist network that encodes a knowledge base of concepts (e.g., objects, actions, plans, and goals) and general knowledge rules for inferencing between concepts, as in ROBIN (e.g. Figure 6). Also indexed into this semantic network are units representing long-term memory episodes that are potential targets for retrieval. Using this network, the understanding and analog retrieval process consists of four major stages:

(1) Activation is spread through the semantic network to disambiguate and infer an interpretation of the cue, as in ROBIN.

(2) Symbolic markers are propagated from the units of the winning inference path to find the targets that are semantically similar in the current context to the cue's interpretation.

(3) A network of units is dynamically built to represent the possible competing mappings between the cue's interpretation and the semantically similar targets found by the spread of markers. The excitatory and inhibitory connections between units of this new mapping network enforce semantic and structural consistency with the cue.

(4) The new mapping network is settled by a constraint-satisfaction process similar to ARCS' that performs competitive retrieval; the mapping units active after settling constitute the most coherent match to the cue.

Because the units in the mapping network formed by the spreading-activation and marker-passing process feed back into the corresponding units in the semantic network, the activation of the target most semantically and structurally similar to the cue increases. The target episode in the semantic network with the highest activation is retrieved.

4.2.1. Cue Disambiguation and Understanding

As previously mentioned, SAARCS is built upon the purely-localist connectionist network of ROBIN (described in Section 3.1). This allows SAARCS to perform lexical and pragmatic disambiguation and reinterpretation, while also being able to represent the variable bindings and perform some of the general knowledge rules necessary for high-level inferencing and understanding.

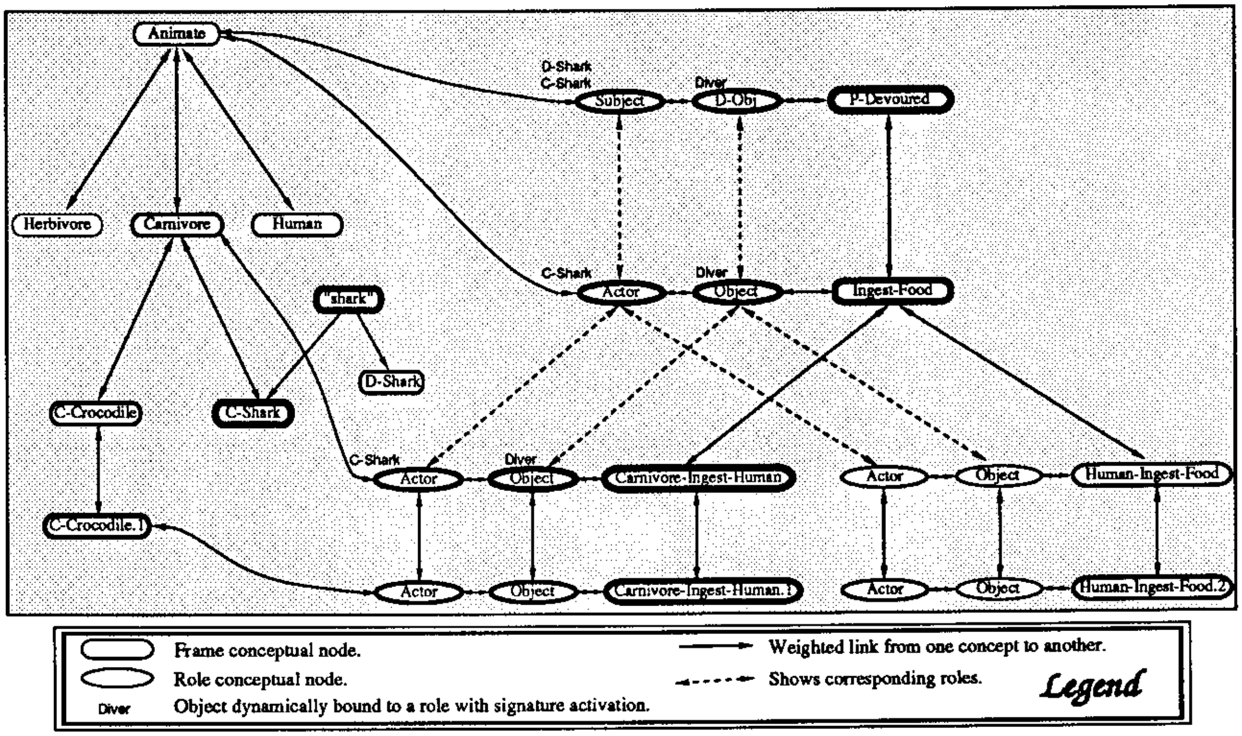

Figure 9. Simplified SAARCS network segment showing some of the conceptual nodes on the evidential layer of the network. The network is similar to that of ROBIN's (e.g. Figures 5a and 5b), except that long-term memory episodes, such as Carnivore-Ingest-Human.1 and Human-Ingest-Food.2, are connected to the network. Shown are the activation of the evidential layer's nodes after presentation of input for *The shark devoured the diver.* The labels next to roles (i.e. D-Shark, C-Shark, and Diver) show the concepts inferred over the binding nodes by propagation of signatures.

In addition to ROBIN's normal network structure encoding frames and rules for inferencing between them with signatures and evidential activation, SAARCS has conceptual units representing the episodes in its long-term memory. Each of the elements of these episodes is an instance of a frame in the semantic network, and so is connected (without signature binding paths) to the evidential units of those frames. The strength of those weights is relative to how "well" the episodes have been remembered: particularly salient episodes will have high connection weights, and "fading" memories will have low connection weights.

Figure 9 shows an example of how a simple episode is connected to the network. This network shows a simplified view of a portion of the evidential units in the network. Shown are the frame for phrase P-Devoured (as in *The shark devoured the diver*), for action Ingest-Food, and two of Ingest-Food's alternative concept refinements, Carnivore-Ingest-Human and Human-Ingest-Food. In Figure 9, the episode *The crocodile ate the swimmer* is represented by the instance Carnivore-Ingest-Human.1, whose Actor is connected to C-Crocodile.1 and whose Object is connected to Swimmer.1 (not shown). *The sailor consumed the fish* is represented by Human-Ingest-Food.2.

To start off the understanding and retrieval process, the input for a cue is presented to the network. Figure 9 shows the results of the spread of activation for the cue *The shark ate the diver*. In this network, the word *shark* has two alternative meaning senses, C-Shark (a large, carnivorous shark) and D-Shark (a cut-up dinner shark). The labels next to the role units in the figure (e.g. C-Shark) represent the bindings inferred by propagation of signatures along paths of binding units like those of Figure 5. The result in Figure 9 shows that the network has disambiguated the word *shark* to the large, carnivorous kind, and inferred that there has been a case of Carnivore-Ingest-Human where a C-Shark Actor has eaten a Diver Object.

4.2.2. Finding Similar Targets

The spread of activation used to understand the input has the side-effect of activating targets that are semantically similar to the interpretation of the cue. For example, in Figure 9, the target Carnivore-Ingest-Human.1 has become strongly activated due to activation from Carnivore-Ingest-Human, as a result of processing input for the sentence *The shark ate the diver* (**Killer Shark**).

To start memory retrieval, symbolic markers are spread from the frame and role units of the cue's winning interpretation. These markers hold both a symbolic backpointer to their originating unit and a strength equal to the numeric product of the connection weights they have propagated over. The frame and role markers only propagate over connections between corresponding frames and roles, respectively, and only over active portions of the network.

This propagation of markers finds, in parallel, all of the instances in memory that are semantically similar to the cue in the current context. Equally important is that the markers' backpointers tell exactly which part of the cue they are similar to. For instance, in Figure 9, one marker will reach Carnivore-Ingest-Human.1 from the inferred Carnivore-Ingest-Human, and another marker will reach C-Crocodile.1 from C-Shark[10].

This marker-passing naturally constrains the search for similar targets because of two features of the network: (1) instances not semantically similar to the cue in the current context will have little or no activation, and so will not be reached (e.g., Herbivore), and (2) instances that are active, but which are semantically distant from a cue concept (such as C-Crocodile.1 from Diver) will not be reached because of their separation in the network.

4.2.3. Building the Mapping Network

The marker-passing process finds large numbers of partially-active long-term instances that are semantically similar to part of the cue. Each of these correspondences is a potential analog. However, to retrieve a single coherent episode most analogous to the cue, these isolated correspondences must compete against each other. This competition is driven by parallel satisfaction of the two main types of constraints, semantic similarity and structural consistency, that are believed to operate in both analogical retrieval [Thagard *et al.*, in press] and analogical mapping [Holyoak & Thagard, 1989].

To perform this competition, a mapping network is dynamically formed whose units represent the possible mappings between each pair of semantically similar concepts, as in the ARCS model of analogical retrieval [Thagard *et al.*, in press]. In SAARCS, these are the pairs found by propagation of the markers. For example, in **Killer Shark**, markers hitting units for the target *The crocodile ate the swimmer* would cause map-

[10]Markers are used here rather than signatures since each instance in the targets may be semantically similar to multiple concepts in larger cue stories, and thus need to hold several markers at once. Since signatures are activation patterns, binding units can hold only one signature binding at a time. Though markers could also be used in place of signatures for the language understanding portion of the network, signatures are used because of the smoother integration of signature activation and selectional restrictions with the activations of the evidential portion of the network (see section 3.2). Future extensions of signatures may make it possible to eventually use them in place of markers throughout SAARCS.

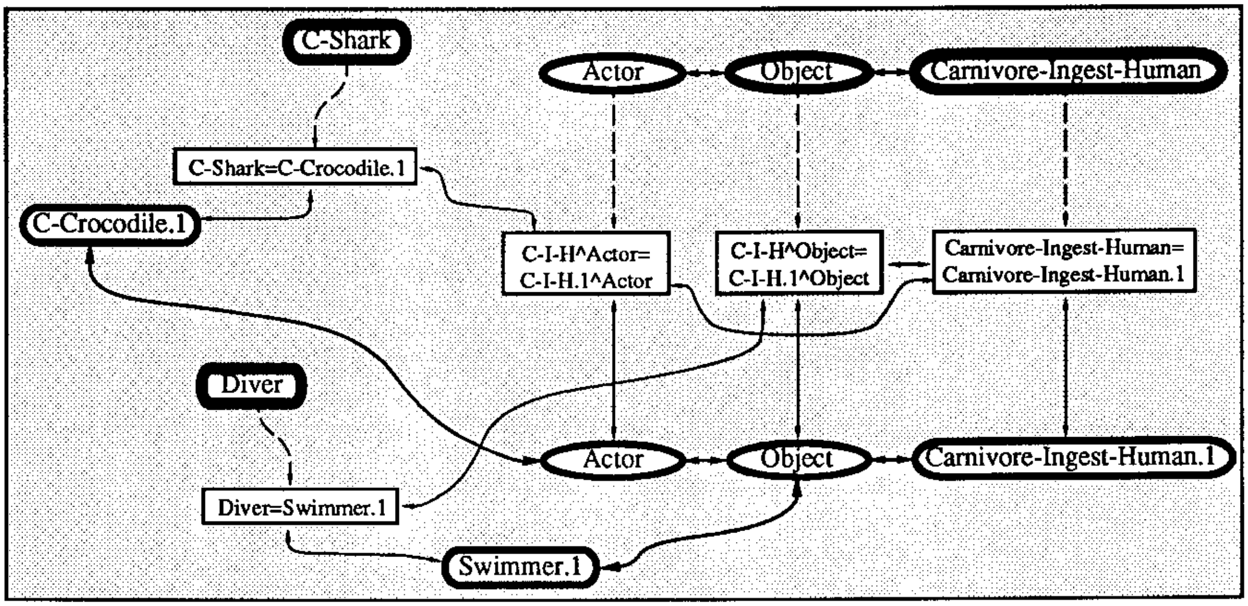

Figure 10. Some of the mapping units (rectangles) created by passing of markers in **Killer Shark**. All links shown are excitatory. Unidirectional dashed lines have weights proportional to the total weight distance between that particular concept and the target being mapped.

ping units to be created for the hypotheses that C-Shark=C-Crocodile.1, Diver=Swimmer.1, and Carnivore-Ingest-Human=Carnivore-Ingest-Human.1. Target roles that receive markers also create units representing the possible mappings between those roles and the markers' originating roles. This in itself enforces partial structural consistency, since only corresponding roles that can be reached over signature inferencing paths (dashed lines in Figure 9) will receive markers. The units created for the potential mappings between **Killer Shark** and *The crocodile ate the swimmer* are shown in Figure 10.

As in ARCS, structural consistency is enforced by excitatory connections between corresponding mapping units. As shown in Figure 10, excitatory connections are created between units mapping two roles (e.g., C-I-H^Actor=C-I-H.1^Actor) and the units mapping their frames (e.g., Carnivore-Ingest-Human=Carnivore-Ingest-Human.1). Units mapping two concepts that serve as the fillers of two mapped roles also have excitatory connections (e.g., between C-Shark=C-Crocodile.1 and C-I-H^Actor=C-I-H.1^Actor).

All of the above types of connections between structurally consistent mapping units have a small positive value (0.05). Excitatory weights are also constructed to mapping units from the units in the semantic network that they map, with the connection weights being proportional to the total path weight product between the concepts (0.05 * strength of the marker that caused the mapping unit to be built). These weights thus give importance to both (a) semantic similarity, since the weights to mapping units for two very similar concepts will be higher than those for two less similar concepts, and (b) pragmatic relevance, since important and relevant goals will have more basic activation in the semantic network, thus biasing retrieval towards units mapping those goals.

Competition between potential mappings is facilitated by inhibitory connections between all rival mappings (-0.20 in the simulation). For instance, there will be an inhibitory connection between Diver=Swimmer.1 and the unit created for Diver=Sailor.2 (from the target *The sailor consumed the fish*).

4.2.4. Competition and Retrieval

During and after creation of the units representing candidate analogical mappings, the new mapping network is settled using a constraint-satisfaction algorithm. The mapping units that are most active after the network has settled will be those that constitute the most coherent match to the cue [Thagard *et al.*, in press].

Because mapping units are created with bi-directional connections from their target units in the semantic network (e.g., from C-Shark=C-Crocodile.1 to C-Crocodile in Figure 10), activation from the winning mappings feeds back into the targets. This boosts the evidential activation of targets most analogous to the cue, so that they tend to become highly activated. The target retrieved is the episode with the highest evidential activation.

4.2.5. Simulation Results

SAARCS has been implemented in the DESCARTES connectionist simulator [Lange *et al.*, 1989]. The model has been tested on three different types of competitive and non-competitive retrieval: examples in which the targets can be retrieved solely on the basis of semantic similarity after interpretation, examples in which analogical similarity plays a crucial role, and examples in which plan/goal analyses of the cue must be made before retrieval is possible.

The first class simulated are retrievals in which a single target is clearly the most semantically similar to the cue after interpretation. In such examples, the interpretation process activates the similar target much more highly than any others. This is the case when *The crocodile ate the swimmer* and *The sailor consumed the fish* are the closest potential targets for *The shark ate the diver*. The carnivorous crocodile episode is so similar to **Killer Shark** that it becomes highly active just from the inferencing process, especially in relation to *The sailor consumed the fish* (see Figure 9). Structural similarity pressures from the mapping network only marginally helps retrieval in these kind of cases.

In other cases, however, multiple targets can have approximately the same semantic similarity to the cue, so structural consistency plays a larger part in retrieval. An example of this is that the target *The sailboat followed the dolphin* (Ptrans-Follow.1) is a better analogy for *The boat followed the whale* than is *The porpoise followed the ferry* (Ptrans-Follow.2). In this kind of case, the pressures due to structural consistency allow the better analogy to be retrieved first.

Figure 11a shows the activations of these target episodes during retrieval in SAARCS. Activation reaches the two targets after presentation of the cue at cycle 16, and the semantic network settles by about cycle 39. Although activations of the cue (not shown) clearly indicate that a Boat was Ptrans-Following a Whale, the activations of the two targets are about the same, essentially because they both involve sea mammals and boats following each other. At this point, markers propagate from the cue's interpretation, so that at cycle 41 the competing mapping units for Ptrans-Follow=Ptrans-Follow.1 and Ptrans-Follow=Ptrans-Follow.2 are formed (Figure 11b). Because of the excitatory connections enforcing structural similarity between them and the other newly created mapping units, Ptrans-Follow=Ptrans-Follow.1 soon begins to win, dominating by about cycle 80. This activation feeds back into the semantic network, driving Ptrans-Follow.1 to saturation and allowing Ptrans-Follow.2 to decay. *The sailboat followed the dolphin* is thus retrieved as the best analogy for the cue.

The final set of simulations have tested SAARCS' ability to perform retrievals which require a plan/goal analysis of the cue. For instance, to understand *John put the pot inside the dishwasher because the police were coming*, the ROBIN portion of the network must first make multiple inferences to decide that John was most likely trying to hide his marijuana from the police inside the dishwasher, because he didn't want to

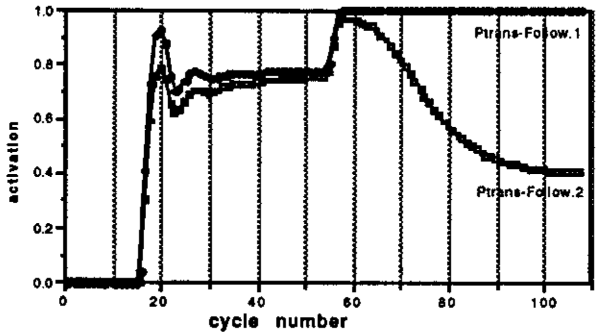

Figure 11a. Activation of target nodes Ptrans-Follow.1 (*The sailboat followed the dolphin*) and Ptrans-Follow.2 (*The porpoises followed the ferry*) after presentation of *The boat followed the whale.*

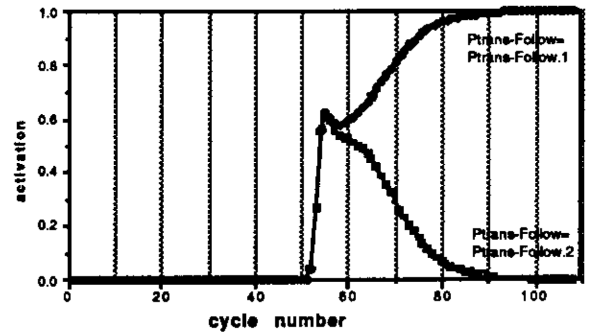

Figure 11b. Activation of mapping units Ptrans-Follow=Ptrans-Follow.1 and Ptrans-Follow=Ptrans-Follow.2 after they were created due to presentation of *The boat followed the whale.*

get arrested. These inferences combined with ROBIN's spread of activation allow the network to disambiguate and form an interpretation of the cue (see Section 3.1).

Once a reinterpretation has been made to Hiding Marijuana due to confluences of evidential activation from the inferred plan/goal analysis, SAARCS uses that inferred interpretation to retrieve the analogous episodes *Bill hid the cocaine in the stove so that he wouldn't be arrested,* as opposed to the previously most analogous episode *Mary put the cooking pot in the dishwasher to clean it.*

4.3. Comparing SAARCS to Non-Hybrid Models

By integrating a localist connectionist network with marker-passing that allows the dynamic creation of a mapping network, SAARCS is able to combine parts of the language understanding and analogical retrieval processes in a network without an external supervisor (other than the one that created the network in the first place). It is thus a first pass at a model explaining the influence of the language understanding on memory retrieval and vice versa. Unlike models that concentrate primarily on memory retrieval, such as symbolic case-based reasoning models (cf. [Schank & Leake, 1989]) and connectionist retrieval models [Thagard *et al.*, in press] [Barnden & Srinivas, in press], SAARCS can potentially account for many psychological phenomena involving priming and language effects in human memory retrieval. These phenomena include increased retrieval due to repetition, recency, and prior semantic priming, all of which can be modelled by variations in evidential activation levels prior to presentation of the cue.

It will be desirable in the future to simplify SAARCS by removing the hybrid portions of the model. For example, it would be desirable to replace the marker-passing that SAARCS uses to find correspondences between semantically similar concepts with a purely spreading-activation approach, such as an extension to signatures. The dynamic creation of mapping units is also quite expensive computationally, so we are looking into ways to instead temporarily recruit pre-existing mapping units. As we found when finding a way to handle some of the variable binding and inferencing abilities of hybrid marker-passing networks in ROBIN, we suspect that finding purely-localist or distributed connectionist techniques to handle the hybrid portions of SAARCS will also proffer a number of advantages, such as learning and more natural integration of understanding and retrieval. For now, however, the hybrid connectionist modelling approach allows us to explore the effect of integration of cognitive functions that have not yet proven amenable to single-level solutions.

5. REMOVING UNREDUCED HYBRID MECHANISMS

Something that often goes unmentioned is that most connectionist models, even those on a single level, are actually hybrid models that rely on human or symbolic intervention to succeed. The simplest example is that the researcher or a program must usually provide networks' inputs and examine the their final outputs to see if they have completed their tasks as desired. This is generally quite reasonable, since connectionist (and artificial intelligence) research is at far too early a stage to build cognitive robots that interact with their environments autonomously.

Somewhat more troublesome is that the human researcher must always specify the structure of the networks in advance. For distributed connectionist networks, this means specifying the numbers of units, their connectivity, the form of the input and output, and various network parameters. For localist and marker-passing networks it is even more difficult, since the researcher must specify the knowledge representation to be encoded by units in the network, how those units are connected, and what the connection weights between units are. However, although specifying the network structure in advance is often tedious, it is also arguably reasonable to work with such networks, on the assumption that some learning or even evolutionary mechanism could be postulated to have come up with that structure in the first place.

Potentially the most problematic human and symbolic interventions in connectionist network processing are those which must often be used *during* network processing. Many models have such hybrid *unreduced mechanisms* that are crucial during network processing but which are not handled by the units and connections of the network. One example is that learning techniques for distributed networks, for the most part, require that all patterns to be learned are known in advance and continually presented to the network as training progresses incrementally. This requires that all of the patterns be stored outside of the network while learning occurs, generally in a symbolic buffer without which learning would fail. Other examples of unreduced mechanisms in connectionist networks are described by Lachter & Bever, [1988] and Aizawa, [this volume].

If connectionist models are to be posited as comprehensive models of cognitive function, it will therefore be necessary to eventually find connectionist implementations of all such hybrid unreduced mechanisms, in much the same way as it is necessary to eventually implement hybrid models on a single connectionist level.

5.1. Short-Term Sequential Memory

One cognitive function that has not yet been implemented in connectionist networks is that of *short-term sequential memory.* Short-term sequential memory is what people use to store short sequences of semantically unrelated items for a period of a few seconds or minutes. Examples of short-term sequential memory are remembering a phone number between the time a telephone operator says it until the time it is dialed, remembering the combination of a lock long enough to copy it down on paper, and so on. People in general are able to remember such ordering information for a maximum of around seven "chunks" of previously established concepts [Miller, 1956].

There have been a few models that implicitly learn and store sequential information in long-term memory, such as the recurrent backpropagation model of [Pollack, in press]. These models, however, store their sequential information by employing time-intensive methods for modifying connection weights, and so are models of long term, rather than short-term sequential memory.

Besides being an important cognitive function in itself to model, short-term sequential memory is one of the unreduced hybrid mechanisms of distributed connectionist networks that deal with time. In order to learn an ordered sequence, a model must be able to store it temporarily. For example, several models have been able to handle and interpret sequential input by using recurrent networks, such as models that take sequential text as input (e.g. [Miikkulainen & Dyer, 1989] , [St. John, 1990], and [Pollack, in press]). The basic training procedure of such models is often to present each element of the sequence in turn as input to the network while training each input (even the first element) to produce the desired output of the entire sequence. Of course, if the models are ever to work without supervision, they will have to have some means to compute the desired training output. Doing this will generally require seeing the complete sequence before training can begin — which means that the actual sequence itself will have to be stored temporarily. Currently that short-term sequential memory can only be done with the hidden hybrid mechanism of a symbolic buffer.

5.2. A Model of Short-Term Sequential Memory

To model short-term sequential memory and potentially eliminate one of the unreduced mechanisms of distributed connectionist networks dealing with time, we have been working on a connectionist model of short-term sequential memory that uses only *activation changes* to temporarily store and recall the exact order of a single, novel short sequence of previously defined concepts [Lange & Allen, 1991].

The basic model consists of a network for *Semantic Memory* and a network of *Responder Groups* (Figure 12). The Semantic Memory holds all of the previously known concepts, and would in theory hold the model's semantic knowledge and rules for use in short-term semantic reasoning. For simplicity, however, the semantic memory is simply a localist winner-take-all network, with a separate unit for each concept. The responder groups serve to temporarily store ordering information of the sequences presented. Each concept in the semantic memory is randomly connected to a subset of the responder groups, which each have a random output threshold.

5.2.1. Storing the Sequence

An ordered sequence is presented to the model for storage by activating, in turn, each of the units in semantic memory representing that element in the sequence. When a given element in the sequence is activated, activation will propagate from it to its responder groups. This new activation will cause a subset of its responder groups to go over threshold and be temporarily *recruited* to represent that element and its position in the sequence. The numbers of responders recruited for each element holds the ordering information.

Figure 12 shows a limited network with ten responder groups and four semantic units (A, B, C, and D). Each of the semantic units is randomly connected to half of the responder groups. The thresholds of the responders have been randomly chosen between 0 and 2. For simplicity, all excitatory connections are of unit weight. We will now present this network with the sequence D - A - C.

To start the sequence, semantic memory unit D is clamped to an activation of 1 (Figure 13a), with all other semantic units having activation 0. Activation spreads from D to each of its potential responders through the responders' input units (bottom layer of the responder groups). Responder 1's activation is now 1, but its threshold is 1.4, so it does not fire. The same is true with Responder 5, whose threshold is 1.5. Responders 4, 8, and 10, however, have thresholds under 1, and so do fire. These responders have been "recruited" to

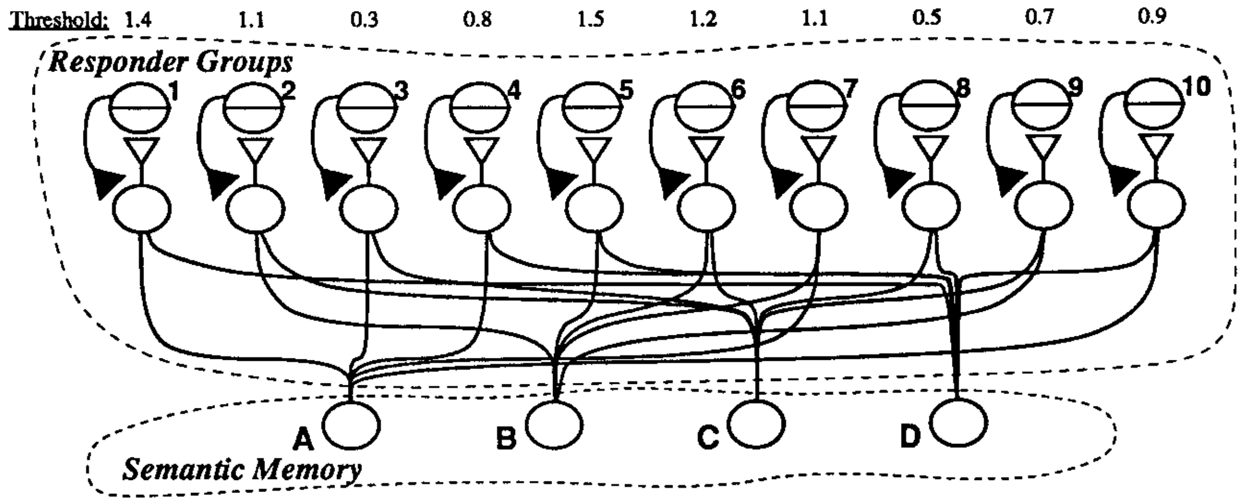

Figure 12. Random interconnections between units in Semantic Memory and Responder Groups. The thresholds on the top line are the thresholds of the top node of each responder group. The links with dark triangles are inhibitory, while the links with white triangles or no triangles are excitatory.

represent the fact that D was the first element in the sequence, with the inhibitory connection to their input units shutting them off from further input.

Unit D is then shut off, and the second unit in the sequence, A, is clamped to an activation of 1 and its activation propagated to its responder groups (Figure 13b). Responder 1, which had an activation of 1 before (from D), now gets enough activation (2 overall) to fire, and is recruited by A. Responder 3 is likewise recruited by A, with Responder 7 gaining activation but not firing. A has no effect on Responders 4 and 10, however, since they have shut themselves off from further input after having been recruited by D. Finally, the third unit in the sequence, C, is clamped, and activation spread (Figure 13c). C is only able to recruit one responder group, Responder 9.

A total of six of the responder groups in Figure 13c were recruited during presentation of sequence D - A - C. The order of the elements in the sequence is implicitly represented by the activation of the responder groups, since three of the responders were recruited by D, two were recruited by A, and one was recruited by C. This ordering information of the network will also occur with other sequences that are presented. For example, if the sequence presented had instead been A - C - D, there would have been (a different) three responders recruited for A (3, 4, and 10), two for C (8 and 9), and one for D (1). As long as the thresholds of the responder groups are randomly set within a certain range, the number of recruited responders for each element will generally vary with its position in the sequence — the first element in the sequence will nearly always recruit the most responder groups, the second element the next most, and so on, as the pool of eligible responder groups becomes smaller for each element in the sequence. This ordering becomes increasingly likely with larger responder group networks.

5.2.2. Retrieving the Sequence

Once a sequence has been presented to the network and each of its elements recruited a decreasing number of responder groups, the sequence can be recalled by feeding activation *back* from the responder groups to the

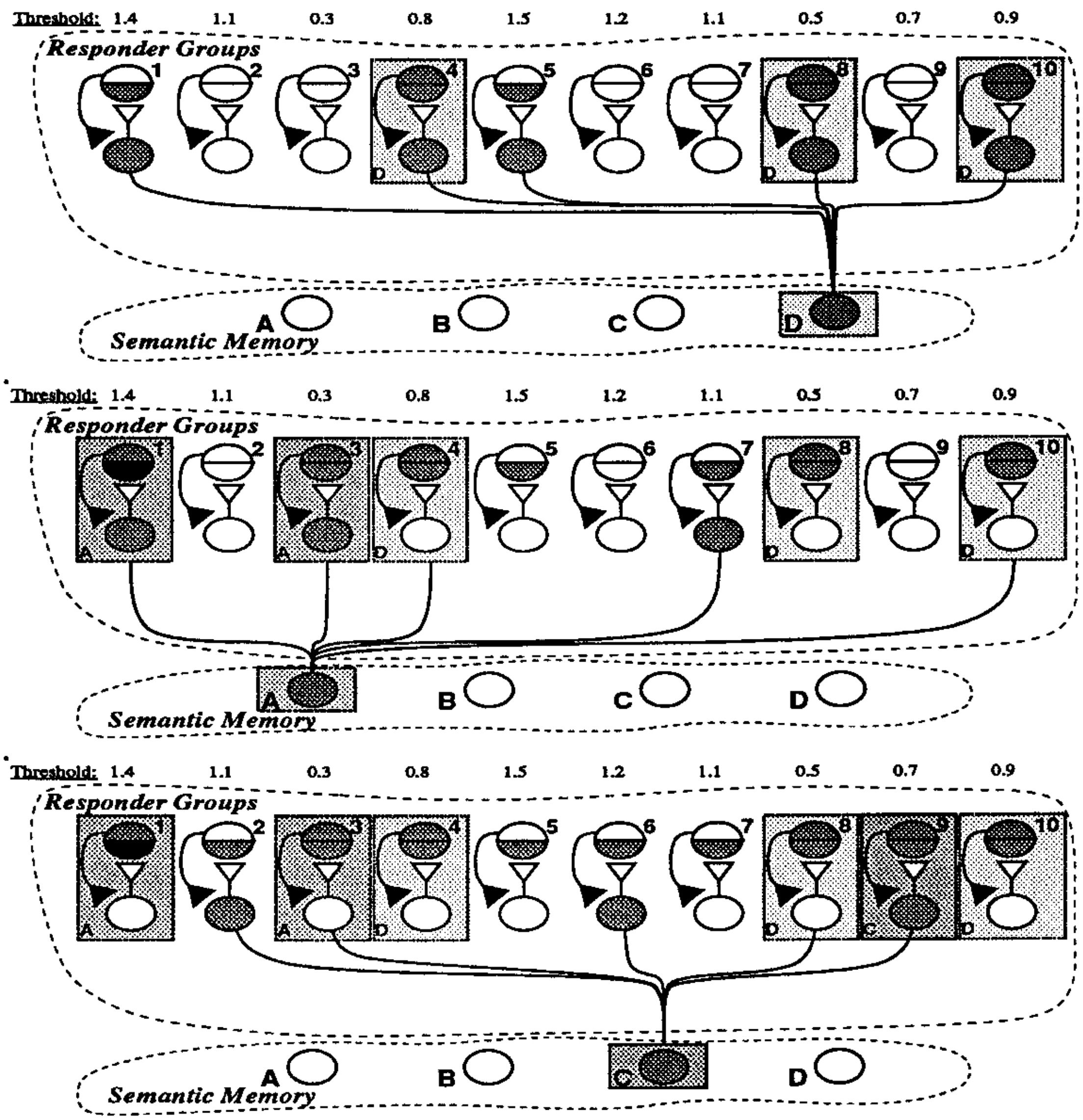

Figure 13. (a) Activation of responder groups after D is activated as the first element of the sequence. The bottom half of the responder nodes shows their level of activation (grey = 1.0), and the top half shows their output (1 if activation greater than threshold, 0 otherwise). Responder groups within the rectangles labeled D have been caused to fire and therefore be "recruited" by the first element in the sequence, D. (b) Activation of responder groups after A has been activated as the second element of the sequence. (c) Activation of responder groups after C has been activated as the third element of the sequence.

units in semantic memory. The first element in the sequence will generally win the competition, because it has connections from all of its recruited responder groups (the largest group of recruited responders).

Unfortunately, such a simple scheme will all too often fail, especially on sequences with repeated elements. If the sequence F - G - G is presented, for example, and F recruits 30 responders, the first G recruits 20,

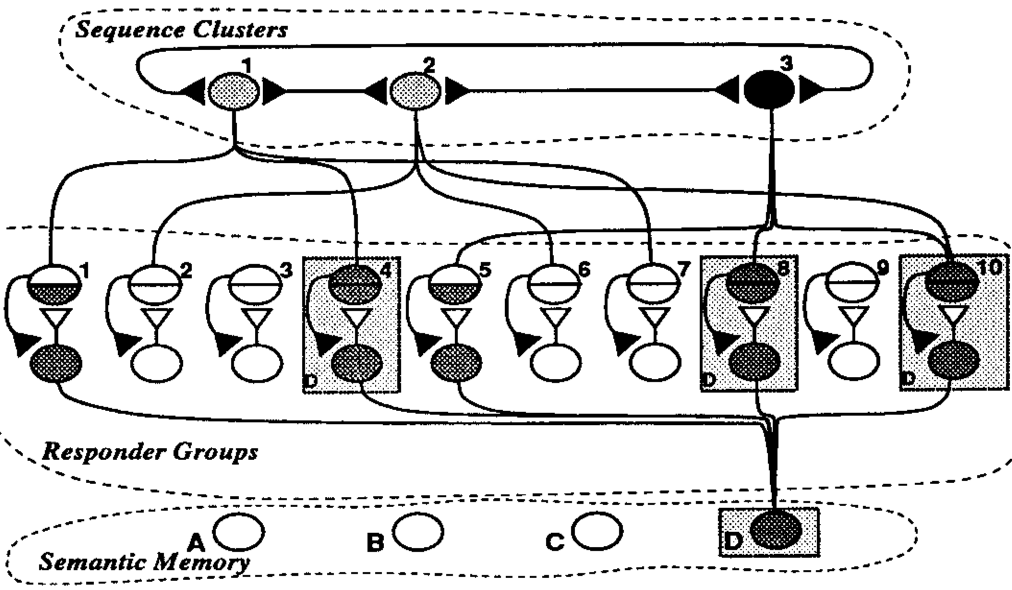

Figure 14. Simplified version of sequence clusters formed in a winner-take-all network and randomly connected to responder groups. Here, Sequence Cluster 3 has inputs from more of the responder groups recruited by D (2) than the other sequence clusters, and so will win the winner-take-all competition and serve to cluster those responders. Sequence Cluster 3 will therefore control their feedback to semantic memory on retrieval.

and the second G recruits 15, then the total number of responders feeding back into G will be 35, causing it to be recalled first.

To handle these kind of retrieval problems, the model has a network of *Sequence Clusters*, each of which is randomly connected to a subset of the responder groups (Figure 14). The newly fired responder groups recruited by the presentation of a given element in the sequence will drive a winner-take-all competition between the sequence clusters. The cluster that best represents the space of newly recruited responder groups (because it happens to have connections to more of the responder groups than any other cluster) will win the competition and cluster those responders — into a group separate from the responders recruited by previous and future elements in the sequence.

At the end of the presentation of the sequence, there will be one sequence cluster active for each element in the sequence. Because there were more newly-recruited responder groups for the first element than for any of the later elements, the cluster representing the first will have input from the largest number of active responders. Similarly, the second cluster will have more active inputs than the third, and so on. An example run, implemented in the DESCARTES connectionist simulator [Lange *et al.*, 1989], is shown in Table 3.

On recall, the active clusters will compete, and the cluster for the first element will win because of its greater number of responder groups (22 in Table 3). This cluster will then feed its activation back to its responder groups and through them down to the semantic memory — causing the first element in the sequence (R) to get the most activation and be recalled. The first cluster then removes itself from the competition (through gating not shown), allowing the cluster representing the second element in the sequence to win and recall the second element. The rest of the sequence is recalled in the same way. The complete sequence can

Element Presented	Responders Recruited	Sequence Cluster #	Responders Clustered
R	36	4	22
B	28	33	18
G	10	29	9
D	8	25	8
S	6	41	5

Table 3. Responder Groups and Sequence Clusters activated when the sequence R - B - G - D - S was presented to a network where each semantic element was randomly connected to 50 of the 100 total responder groups. The network had 50 sequence clusters, each of which had random connections to 50 of the responder groups. The first column shows the element presented, the second column shows the number of responder groups recruited by that element, the third column shows which sequence cluster won to represent that element, and the fourth column shows how many of the responders that cluster actually represents.

be recalled repeatedly, until a new sequence is stored or all of the activation in the responder groups decays away.

5.3. On Removing Hidden Hybrid Mechanisms

The model we have just described is in the early stages of testing. As a model of human short-term sequential memory, it currently has several shortcomings. For example, because there are fewer and fewer responder groups available as elements are presented, it tends to remember the first elements of a sequence the best, and remember the last elements the worst. People, however, are worst at repeating elements in the middle of a sequence, and not the end [Miller, 1956]. We plan to experiment with various changes in decay rates and the basic architecture to see if we can more closely match such psychological data.

On the other hand, this model seems successful as a first pass at modelling short-term sequential memory in a distributed connectionist network, and therefore as a first step towards removing the hybrid symbolic buffer mechanism from long-term memory distributed models that operate on sequential data.

6. CONCLUSIONS

The distributed connectionist, localist connectionist, and marker-passing levels of connectionist processing each have a different set of strengths and weakness. There is a large gap between the low-level areas that subsymbolic (distributed connectionist) models are well-suited to modelling and the high-level areas that symbolic (marker-passing and traditional AI) models are. Because of this, it is often impossible to build a model of a given cognitive task solely from elements of a single connectionist level. The only possible solution in these cases is often to build hybrid models that combine elements and capabilities from multiple connectionist levels.

Besides allowing progress in problems that cannot be handled otherwise, hybrid connectionist models serve as useful guides to potentially valuable areas of future research in single connectionist levels. If a hybrid model is successful, then it serves as strong evidence that an attempt to map the abilities of the hybrid model onto a single level might be a good way to solve the problem. On the other hand, if the hybrid model is unsuccessful, then the attempt shows that either a different approach is needed or that there are more facets to the problem than originally expected.

Of course, if hybrid connectionist models are to be of value to researchers ultimately interested in building models on a single connectionist level, then it must eventually be possible to at least roughly map the hybrid capabilities into that level. It is therefore preferable to use mechanisms that are as similar to each other as possible in building hybrid models to increase their chance of being implemented in a single level. When a hybrid model is actually mapped to a single connectionist level, it will likely have advantages over the hybrid because of better integration with the rest of the level than the sometimes awkward integration between elements in hybrid models. It is also important to note that *most* connectionist models are actually hybrids; even single-level models often use symbolic mechanisms (such as temporary storage buffers) to allow processing or training to succeed. These unreduced hybrid mechanisms must also be eventually mapped to units and connections in the connectionist level.

Our research has resulted in three models that illustrate the value of the hybrid connectionist modelling approach:

(1) SAARCS, a hybrid localist connectionist/marker-passing model that is able to integrate aspects of the language understanding process with the analogical retrieval process. Because of the hybrid capabilities it uses, it is able to model the influence of comprehension on retrieval in a way that single-level connectionist or symbolic models have not yet been able to, including potentially being able to account for many psychological phenomena involving priming on human memory retrieval.

(2) ROBIN, a purely-localist connectionist network that has many of the variable binding and inferencing abilities of hybrid marker-passing/localist connectionist models. ROBIN illustrates that the abilities of hybrid models can sometimes be mapped onto a single connectionist level, with the resulting model gaining advantages in both simplicity and processing abilities.

(3) A purely-distributed connectionist model of short-term sequential memory that is a start towards mapping one of the unreduced hybrid mechanisms necessary for training of sequential distributed connectionist networks.

REFERENCES

Aizawa, K. (in press): Rumelhart and Norman's Methodology for Connectionism. In J. Dinsmore (ed.), *Closing the Gap: Symbolic vs. Subsymbolic Cognitive Processing.*

Ajjanagadde, V. & Shastri, L. (1989): Efficient Inference With Multi-Place Predicates and Variables in a Connectionist System. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.* Ann Arbor, MI, August 1989.

Barnden, J. (1990): The Power of Some Unusual Connectionist Data-Structuring Techniques. In J. A. Barnden and J. B. Pollack (eds.), *Advances in Connectionist and Neural Computation Theory.* Norwood, NJ: Ablex.

Barnden, J., & Srinivas, K. (in press): Overcoming Rule-Based Rigidity and Connectionist Limitations Through Massively-Parallel Case-Based Reasoning. *International Journal of Man-Machine Studies.*

Charniak, E. (1986): A Neat Theory of Marker Passing. *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86).* Philadelphia, PA, August, 1986.

Diederich, J. (1990): Steps Toward Knowledge-Intensive Connectionist Learning. In J. Barnden and J. Pollack (eds.), *Advances in Connectionist and Neural Computation Theory.* Norwood, NJ: Ablex.

Dolan, C. P. & Smolensky, P. (1989): Tensor Product Production System: A Modular Architecture and Representation. *Connection Science, 1 (1)*, p. 53-68.

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

Dyer, M. G. (1990): Symbolic Neuroengineering for Natural Language Processing: A Multi-Level Research Approach. In J. Barnden and J. Pollack (eds.), *Advances in Connectionist and Neural Computation Theory*. Norwood, NJ: Ablex.

Eiselt, K. P. (1987): Recovering From Erroneous Inferences. *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*. Seattle, WA, July 1987.

Feldman, J. A. & Ballard, D. H. (1982): Connectionist Models and Their Properties. *Cognitive Science, 6 (3)*, p. 205-254.

Fodor, J. A. & Pylyshyn, Z. W. (1988): Connectionism and Cognitive Architecture: A Critical Analysis. In S. Pinker and J. Mehler (eds.), *Connections and Symbols*, p. 3-71. Cambridge, MA: The MIT Press.

Fukushima, K., Miyake, S., & Ito, T. (1983): Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition. *IEEE Transactions on Systems, Man, & Cybernetics, SMC-13, 5*, p. 826-834.

Gasser, M. (1988): *A Connectionist Model of Sentence Generation in a First and Second Language*. Technical Report No. UCLA-AI-88-13 (Doctoral Dissertation), Computer Science Department, University of California, Los Angeles.

Granger R. H., Eiselt, K. P., & Holbrook, J. K. (1986): Parsing with Parallelism: A Spreading Activation Model of Inference Processing During Text Understanding. In J. Kolodner and C. Riesbeck (eds.), *Experience, Memory, and Reasoning*, p. 227-246. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Hendler, J. (1988): *Integrating Marker-Passing and Problem Solving: A Spreading Activation Approach to Improved Choice in Planning*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Hendler, J. (1989a): Editorial: On the Need for Hybrid Systems. *Connection Science, 1 (3)*, p. 227-229.

Hendler, J. (1989b): Marker-Passing Over Microfeatures: Towards a Hybrid Symbolic/Connectionist Model. *Cognitive Science, 13 (1)*, p. 79-106.

Holldobler, S. (1990): A Structured Connectionist Unification Algorithm. *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-90)*. Boston, MA, July 1990.

Holyoak, K. J. (in press): Symbolic Connectionism: Toward Third-Generation Theories of Expertise. To appear in K. A. Ericsson & J. Smith (eds.), *Towards a General Theory of Expertise: Prospects and Limits*. Cambridge, MA: Cambridge University Press.

Holyoak, K. J., & Koh, K. (1987): Surface and Structural Similarity in Analogical Transfer. *Memory & Cognition, 15*, 332-340.

Holyoak, K. J., & Thagard, P. (1989): Analogical Mapping by Constraint Satisfaction. *Cognitive Science, 13*, 295-355.

Kintsch, W. (1988): The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Review, 95*, 163-182.

Kitano, H., Tomabechi, H. & Levin, L. (1989): Ambiguity Resolution in DMTRANS PLUS. *Proceedings of the Fourth Conference of the European Chapter of the Association of Computational Linguistics*. Manchester University Press.

Kwasny, S. C. & Faisal, K. A. (in press): Symbolic Parsing Via Sub-Symbolic Rules. In J. Dinsmore (ed.), *Closing the Gap: Symbolic vs. Subsymbolic Cognitive Processing.*

Lachter, J. & Bever, T. G. (1988): The Relation Between Linguistic Structure and Associative Theories of Language Learning — A Constructive Critique of Some Connectionist Learning Models. In S. Pinker and J. Mehler (eds.), *Connections and Symbols*, p. 195-247. Cambridge, MA: The MIT Press.

Lange, T. (in press): Lexical and Pragmatic Disambiguation and Reinterpretation in Connectionist Networks. *International Journal of Man-Machine Studies.*

Lange, T. & Allen, C. (1991): *A Connectionist Model of Short-Term Sequential Memory.* Unpublished manuscript. Computer Science Department, University of California, Los Angeles.

Lange, T. & Dyer, M. G. (1988): Dynamic, Non-Local Role-Bindings and Inferencing in a Localist Network for Natural Language Understanding. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems I*, p. 545-552. San Mateo, CA: Morgan Kaufmann (Collected papers of the IEEE Conference on Neural Information Processing Systems — Natural and Synthetic, Denver, CO, November 1988).

Lange, T. & Dyer, M. G. (1989): High-Level Inferencing in a Connectionist Network. *Connection Science, 1 (2)*, p. 181-217.

Lange, T., Hodges, J. B., Fuenmayor, M., & Belyaev, L. (1989): DESCARTES: Development Environment For Simulating Hybrid Connectionist Architectures. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society (CogSci-89).* Ann Arbor, MI, August 1989.

Lange, T., Melz, E., Wharton, C., & Holyoak, K. (1990): Analogical Retrieval Within a Hybrid Spreading-Activation Network. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton (eds.), *Proceedings of the 1990 Connectionist Models Summer School.* San Mateo, CA: Morgan Kaufmann.

Lee, G., Flowers, M., & Dyer, M. G. (1990): Learning Distributed Representations of Conceptual Knowledge and Their Application to Script-Based Story Processing. *Connection Science, 2 (4)*, p. 313-345.

McClelland, J. L. & Kawamoto, A. H. (1986): Mechanisms of Sentence Processing: Assigning Roles to Constituents of Sentences. In McClelland & Rumelhart (eds.), *Parallel Distributed Processing: Vol 2*, p. 272-325. Cambridge, MA: The MIT Press.

Miikkulainen, R. & Dyer, M. G. (1989): A Modular Neural Network Architecture for Sequential Paraphrasing of Script-Based Stories. *Proceedings of the International Joint Conference on Neural Networks.* Washington, DC, June, 1989.

Miller, G. A. (1956): The Magical Number Seven, Plus or Minus Two: Some Limits On Our Capacity for Processing Information. *Psychological Review, 63 (2)*, p. 81-97.

Nickerson, R.S. (1984): Retrieval Inhibition From Part-Set Cuing: A Persisting Enigma in Memory Research. *Memory & Cognition, 12*, p. 531-552.

Norvig, P. (1989): Marker Passing as a Weak Method for Text Inferencing. *Cognitive Science, 13 (4)*, p. 569-620.

Pazzani, M. J., Dyer, M. G., & Flowers, M. (1987): Using Prior Learning to Facilitate the Learning of New Causal Theories. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87).* Milan, Italy, August 1987.

Pinker, S. & Prince, A. (1988): On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition. In S. Pinker and J. Mehler (eds.), *Connections and Symbols*, p. 73-193. Cambridge, MA: The MIT Press.

Pollack, J. B. (in press): Recursive Distributed Representations. *Artificial Intelligence*.

Riesbeck, C. K. & Martin, C. E. (1986): Direct Memory Access Parsing. In J. Kolodner and C. Riesbeck (eds.), *Experience, Memory, and Reasoning*, p. 209-226. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Rose (1990): Appropriate Uses of Hybrid Systems. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton (eds.), *Proceedings of the 1990 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

Ross, B. H. (1989): Distinguishing Types of Superficial Similarities: Different Effects on the Access and Use of Earlier Problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 456-468.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986): Learning Internal Representations by Error Propagation. In Rumelhart & McClelland (eds.), *Parallel Distributed Processing: Vol 1*, p. 318-364. Cambridge, MA: The MIT Press.

Rumelhart, D. E. & McClelland, J. L (1986): On Learning the Past Tenses of English Verbs. In McClelland & Rumelhart (eds.), *Parallel Distributed Processing: Vol 2*, p. 216-271. Cambridge, MA: The MIT Press.

Schank, R. C., & Leake, D. B. (1989): Creativity and Learning in a Case-Based Explainer. *Artificial Intelligence, 40*, 353-385.

Shastri, L. (1988): A Connectionist Approach to Knowledge Representation and Limited Inference. *Cognitive Science, 12 (3)*, p. 331-392.

St. John, M. F. (1990): *The Story Gestalt: Text Comprehension by Cue-based Constraint Satisfaction*. Technical Report No. 9004 (Doctoral Dissertation), Department of Cognitive Science, University of California, San Diego.

Sumida, R. A. & Dyer, M. G. (1989): Storing and Generalizing Multiple Instances While Maintaining Knowledge-Level Parallelism. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*. Detroit, MI, August 1989.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (in press): Analog Retrieval by Constraint Satisfaction. *Artificial Intelligence*.

Touretzky, D. S. (1990): Connectionism and Compositional Semantics. In J. A. Barnden and J. B. Pollack (eds.), *Advances in Connectionist and Neural Computation Theory*. Norwood, NJ: Ablex.

Touretzky, D. S. & Hinton, G. E. (1988): A Distributed Connectionist Production System. *Cognitive Science, 12 (3)*, p. 423-466.

Waltz, D. & Pollack, J. (1985): Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. *Cognitive Science, 9 (1)*, p. 51-74.

Wermter, S. & Lehnert, W. G. (1989): A Hybrid Symbolic/Connectionist Model for Noun Phrase Understanding. *Connection Science, 1 (3)*, p. 255-272.