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

REMIND: RETRIEVAL FROM EPISODIC MEMORY BY INFERENCING AND DISAMBIGUATION

Trent E. Lange
Charles M. Wharton

October 1992
CSD-920047

REMIND: Retrieval From Episodic Memory by INferencing and Disambiguation[1]

Trent E. Lange
Artificial Intelligence Laboratory
Department of Computer Science
University of California, Los Angeles

Charles M. Wharton
Department of Psychology
University of California, Los Angeles

Abstract

Most AI simulations have modeled memory retrieval separately from language understanding even though both activities seem to use many of the same processes. This paper describes REMIND (Retrieval from Episodic Memory through INferencing and Disambiguation), a structured spreading-activation model of integrated text comprehension and episodic reminding. In REMIND, activation is spread through a semantic network that performs dynamic inferencing and disambiguation to infer a conceptual representation of an input cue. Because stored episodes are associated with concepts used to understand them, the spreading-activation process also activates any memory episodes in the network that share features or knowledge structures with the cue. After the cue's conceptual representation is formed, the network recalls the memory episode having the highest activation. Since the inferences made from a cue often include actors plans and goals only implied in a cue's text, REMIND is able to get abstract, analogical remindings that would not be possible without an integrated understanding and retrieval model.

---

[1]This paper will appear in J. Barnden and K. Holyoak (Eds.), *Advances in connectionist and neural computation theory, volume II: Analogical connections*. Norwood, NJ: Ablex.

1. Introduction

The most parsimonious account of language comprehension and episodic reminding is that they amount to different views of the same mechanism (Schank, 1982, p. 23). Consider the following:

> *There were sightings of Great Whites off Newport, but Jeff wasn't concerned. The surfer was eaten by the fish. They found his board with a big chunk cut out.* (**Killer Shark**)

When reading this passage, we may be reminded of analogous stories of people being eaten by sharks, or, more abstractly, of others who knowingly ventured into mortal danger and suffered the consequences (e.g., skiers being buried under avalanches they were warned about). Why do these remindings occur? To comprehend this passage, a reader must find structures in memory that will provide important inferred information, such as the goals and plans of story characters and the characteristic features of events and locations. Thus, in the process of constructing the meaning of a text passage, we may be reminded of similar episodes because these episodes were understood with (and have became associated with) the same knowledge structures.

In spite of the apparent connectedness of comprehension and memory, artificial intelligence simulations of memory retrieval have usually modeled reminding separately from story and language understanding. While this approach may make accounts of each phenomena more manageable, it is undeniable that real-world retrieval cues are the product of the comprehension process. Further, the manner in which a fully elaborated scene interpretation, or discourse model (Kintsch, 1988), is constructed from an explicit textual representation will influence what is retrieved from memory. Thus, we believe that a model that integrates the process by which a cue is understood with the process by which it is used to recall information from memory can make an important contribution to the understanding of episodic memory retrieval.

In this chapter we describe REMIND (Retrieval from Episodic Memory through INferencing and Disambiguation), a structured connectionist spreading-activation model that integrates language understanding and memory retrieval. We start by giving an overview of the comprehension and reminding processes that REMIND models. We then summarize relevant psychological and artificial intelligence (AI) investigations of sentential comprehension, word sense selection, inference generation, and episodic reminding. Next, we describe and demonstrate how the model performs language understanding and memory retrieval. We conclude by showing several examples to illustrate REMIND's inferencing and reminding abilities and compare them with those of other AI and psychological models of episodic reminding.

1.1 Aspects of Reminding

All researchers agree that people tend to be reminded of episodes from memory that are somehow similar to a cue story or thought, as opposed to being reminded of episodes that are completely dissimilar. However, debate has concerned whether memory retrieval is affected by surface similarity, thematic similarity or a combination of both (cf. Seifert, in press). Surface, or *superficial*, similarity between a cue story and an episode occurs when both share similar features, such as similar actors, places, or actions. Thematic, or *analogical*, similarity occurs when episode representations mostly share the same abstract goals, plans, roles, causal structures, belief, and attitudes. It is generally assumed that thematic remindings are more useful than surface remindings for a problem-solver because thematically similar remindings are likely to contain information most relevant to the problem at hand. Different computational models of memory retrieval have made use of surface and abstract similarities to varying degrees, but empirical evidence from cognitive psychology indicates that both types of similarities have an important effect on the episodes of which people are reminded.

There is overwhelming psychological evidence that human memory retrieval is highly sensitive to the degree of surface feature overlap between the cue and long-term memory episodes (Gentner & Landers, 1985; Ratterman & Gentner, 1987; see discussion in Gentner, 1989, pp. 226-233). In the case of **Killer Shark**, a person would likely be reminded of other stories that also involve Great Whites, surfers, surf boards, Newport, or eating, because these stories contain concepts that are semantically associated to individual lexical items in this story. In general, people tend to recall stories that have a large semantic overlap with what they are currently thinking about.

While the influence of surface similarity on remindings has generally been agreed upon, the effect of thematic similarity on memory retrieval is still a matter for debate. The most robust finding in the analogy literature is that people often fail to retrieve analogous, but superficially dissimilar, targets (e.g., Gick & Holyoak, 1980; Keane, 1988; Ross, 1987; Seifert, McKoon, Abelson, & Ratcliff, 1986; Spencer & Weisberg, 1986). Gentner and her associates (Gentner & Landers, 1985; Ratterman & Gentner, 1987) have found reliable retrieval advantages for cue/target story pairs that shared similar concrete nouns but no analogical similarity in comparison with story pairs that, conversely, were analogically similar but shared no similar concrete nouns. In the context of their SME model of analogical mapping, these findings led Falkenhainer, Forbus, and Gentner (1989, p. 35) to propose that memory access is determined by object-attribute similarity (mere-appearance rules), but not the relational similarity between cues and targets.

Intuitively, however, using only the surface features of a text does not seem to tell the whole story of memory retrieval. In fact, it misses most of the story — the actual meaning that a person infers to understand the surface features. Using only the surface features for memory retrieval in **Killer Shark**, for example, would miss the important inferences that the surfer took his surfboard out into the waters off Newport, that he did so despite being warned of the danger, and that he was therefore killed in a vicious fashion by a shark. These inferences would seem to be at least as important cues as surface features such as "sightings", "Great White", and "eaten". Even more valuable, from a planning perspective, is the ability to access episodes by more abstract features inferred from the text, so that the memories can be useful in other contexts. For example, recall of the planning failure caused by ignoring the danger warning in **Killer Shark** could literally be life-saving to a person who was about to ignore another life-threatening situation (such as warning of avalanches).

Such an approach is taken by many AI models of episodic reminding (e.g., Kolodner, 1984) and case-based reasoning (e.g., Hammond, 1989; Kolodner, Simpson, & Sycara, 1985; Owens, 1989; Schank & Leake, 1989). Because of their problem-solving orientation, such models generally attempt to retrieve the episode from memory that is most likely to help them in their current task. To do this, they search memory using intelligent indexing methods for the best-matching episodes that share an analogous structure of goals, plans, enablements, or failures with the current problem situation, depending on the reasoning task. Almost all case-based reasoning models use highly structured representations of episodes (or cases) that include not only their surface features, but also abstract features and structures that allow them to be retrieved at useful times.

Case-based reasoning models have received indirect support from several psychological experiments. For example, the notion that inferred explanations for the reason something failed will cause people to think of other episodes with similar failures was tested by Read and Cesa (1990). These authors found that when subjects were asked to give reasons for unexpected events in stories, they were reminded of previously read stories that had analogous outcomes. Similarly, in contrast to the claims of Falkenhainer et al. (1989), several empirical studies have shown that reminding is sensitive to cue/target analogical or thematic similarity at least in some cases (Gentner & Landers, 1985; Holyoak & Koh, 1987; Ross, 1989; Wharton, Holyoak, Downing, Lange, & Wickens, 1991, 1992; Gick & McGarry, in press). In two experiments in Wharton et al. (1991), subjects studied a number of pairs of two competing passages, such as **Killer Shark** and the following:

> *Larry had never had sushi before. His friends bet $20 he couldn't eat everything on the plate, but they lost. The sailor consumed the fish.*

The individual words of the competing passages were equally associated with the reminding cues (e.g., *The diver devoured the eel*), but only one of the competing passages shared an analogous sentence with the cue (e.g., *The sailor consumed the fish*). Wharton et al. (1991) found that analogous passages were recalled more frequently than disanalogous passages when surface similarities were equated, especially when there was more than one related story in memory. In an experiment having a similar competitive design, Wharton et al. (1992) found that target stories sharing the same abstract theme with a cue were more likely to be retrieved than competing stories that had equivalent surface similarities but which were not thematically related to the cue.

In general, psychological evidence seems to support a theory of reminding in which surface similarities between a cue and the target episodes in memory form a major basis for retrieval, but for which deeper structural similarities and abstract meanings inferred during the understanding and planning process also play an important part.

1.2 The Need For Integrating Reminding and Understanding

As mentioned previously, most psychological and artificial intelligence models of memory retrieval and language understanding have looked at the processes of comprehension and retrieval in isolation. Due to the enormous complexity of human memory retrieval and language comprehension and the limited understanding that we have of them, such an approach has been necessary to make any progress at all. Models of analogical retrieval and case-based retrieval are usually given a complete representation of the input cues, a representation that is either explicitly or implicitly assumed to be the result of general comprehension or reasoning processes. Similarly, models of language understanding (cf. Schank & Abelson, 1977; Dyer, 1983; Kintsch, 1988; Norvig, 1989) have generally performed only language understanding, and not episodic retrieval. The exceptions that perform both (e.g., Lebowitz, 1980; Kolodner, 1984) have generally implemented the understanding and reminding processes as relatively separate modules using a conventional language parser to understand the story and passing its output representation to the reminding model as a cue.

We believe that building an *integrated* model of language understanding and episodic memory retrieval will allow insights into the processes of both that cannot be gained by modeling them separately. The relationship between language comprehension and episodic reminding is neither simple nor unidirectional. Not only does reading or hearing something occasionally cause one to be reminded of similar episodes, but those remindings themselves can have an important effect on subsequent reasoning and comprehension. An integrated model is perhaps the only way to successfully model some of the more subtle aspects of this interaction. One example is that almost all English language words have multiple senses (e.g., river *bank*, money *bank*). Given the previously noted importance of cue/target lexical similarity, accounting for how individual words are semantically disambiguated is important for a theory of reminding. If the ambiguous sentence *John shot some bucks* is read in a context in which a forest is being talked about, a likely interpretation is that John shot a few deer with a gun (Waltz & Pollack, 1985). However, if the same sentence is read in a context involving casinos, a more likely interpretation is that John lost some money while gambling. Although it has yet to be demonstrated empirically, the different interpretations reached by language understanders in different contexts should also affect what episodes they recall. In the first context, one would likely be reminded of episodes involving hunting, whereas in the second context, one would be likely to be reminded of episodes involving gambling — an obvious effect of the understanding process on reminding. More telling, however, is that the stories one is reminded of themselves affect the context in which subsequent sentences are read. For example, the context of the forest itself could have been activated by having just recalled a particular trip to a national forest after a discussion about vacations. Such priming effects could only be explained by an integrated model in which the understanding and reminding processes interact with and directly affect each other.

Finally, modeling episodic reminding and discourse understanding within an integrated model imposes important constraints on the types of processing and knowledge that can be used in either mechanism. If reminding cues are the direct result of the language comprehension process, then the type of indices that those cues contain is limited to the information that the normal understanding process can (and does) infer. Without modeling this integration and thereby constraining the representations used as recall cues, reminding models are in danger of using information (or input) that might not normally be available for retrieval. For example, some case-based reasoning models routinely assume that the representations of cue stories contain high-level thematic inferences. This is because such inferences are necessary for getting the cross-contextual remindings that these models need for analogical transfer and problem-solving to take place. However, it is not necessarily reasonable to assume that people always recognize the high-level themes in the stories they read. This seemed to have been the case in a study by Seifert et al. (1986) in which subjects read two stories that were either superficially dissimilar instantiations of the same theme (such as closing the barn door after the horse has gotten out), or instantiations of two different themes. When subjects simply read these stories, there was no evidence for cross-contextual reminding in a speeded recognition task. However, when the subjects were instructed to pay special attention to the thematic structure of the target stories (and therefore were presumably more likely to recognize or infer their abstract themes), an effect of the similar thematic structure on reminding was found. Only by building an integrated model of language comprehension and reminding can one expect to model the specific circumstances under which understanders infer and can use thematic information in probing memory.

1.3 Overview of REMIND

To explore how comprehension and reminding processes interact, we have developed REMIND, a spreading-activation model that integrates language understanding and memory retrieval. REMIND is initially given a syntactic representation of a short input text as a memory cue. Using general knowledge stored in its long-term memory, REMIND constructs an elaborated interpretation of the cue, and then retrieves the sentence or episode that is most similar to the surface and inferred features of that interpretation. REMIND is a model of the type of deliberate, non-accidental reminding that would occur when one intentionally uses a cue to probe memory, as in attempting to remember an analogous solution to a current problem. While REMIND does not currently model unintentional memory reminding, we are optimistic that much of what we have developed will generalize to such a theory.

REMIND's structured spreading-activation networks encode world knowledge about concepts and general knowledge rules for inferencing in the same way as ROBIN (Lange & Dyer, 1989; Lange, 1992), a structured connectionist model that performs some of the high-level inferencing and disambiguation processes needed for natural language understanding. REMIND's networks also contain representations of prior episodes, such as *Fred put his car in the car wash before his date with Wilma* (**Car Wash**) and *Billy put his Playboy under the bed so his mother wouldn't see it and spank him* (**Dirty Magazine**). The representations of these episodes are the actual plan/goal analysis (or discourse model) that was inferred by the network when input for them was first presented to the network to be understood. These prior episodes are indexed into the semantic comprehension network through connections with all the knowledge structures with which they were understood.

To perform retrieval, REMIND is given a short text passage to use as a deliberate memory cue, such as *John put the pot inside the dishwasher because company was coming* (**Dinner Party**). Units in the network representing the cue and its syntactic bindings are clamped to high levels of activation, which then spreads through the network. By propagating *signature* activation, the network makes the different possible inferences that might explain the input (Lange & Dyer, 1989; see section 3 below). For example, one of the multiple interpretation paths that gets inferred as a possible explanation for John putting the pot inside the dishwasher in **Dinner Party** is that John was trying to clean the pot to satisfy his goal of having everything ready for entertaining his guests. Other interpretations concurrently activated include the possibilities that he was trying to store the pot or hide it. Activation spreads until the network settles and the units representing the most plausible set of inferences has the most activation. The final most highly-activated chain of inferences represents the network's disambiguated plan/goal interpretation of the cue.

Because the units representing long-term memory episodes are connected within the network, an important side-effect of the understanding process is that episodes having concepts related to the elaborated cue also become highly activated. This includes episodes related because there is superficial semantic overlap with the cue (e.g., episodes involving other kitchen appliances or guests) and episodes related abstractly because they share similar inferred plans and goals of their actors (e.g., the **Car Wash** episode becomes activated after receiving the **Dinner Party** cue because both share the inferences that a person was trying to Clean something in preparation for an Entertainment act). After the network settles, the episode that received the most activation from the cue's interpretation and surrounding context becomes the most highly activated, and is therefore retrieved as the best match for the cue.

REMIND is thus an integrated model in which a single mechanism drives both the language understanding and memory retrieval processes. The same spreading-activation mechanism that infers a single coherent interpretation of a cue also activates the episodes the model retrieves from memory. Activation of these episodes combines evidence from both the surface semantics of the input (i.e., different possible word and phrase meanings) and the deeper thematic inferences made from the input, so that the recalled episodes depend on both surface and analogical similarities with the cue. Further, because all representations of the cue and target episodes used in memory retrieval are constructed from inferences made by the language understanding portion of the model, REMIND predicts that the ability to recall analogous episodes directly depends on the context and level of processing when the input was originally understood. And finally, because both inferencing and memory retrieval occur within a single integrated network, the context in which interpretations are formed effects the episodes that are retrieved, which in turn influence the context in which disambiguation and interpretation of input cues takes place. Thus, text comprehension and memory retrieval processes are tightly coupled and strongly effect each other.

Table 1: Types of inferences needed to understand the sentence *John put the pot inside the dishwasher because the police were coming.* (Hiding Pot).

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.

1.4 Language Understanding and High-Level Inferencing

The part of the natural language understanding process that REMIND concentrates on is the problem of *high-level inferencing* (Lange & Dyer, 1989). Because everything that REMIND infers becomes part of the representation of the cue, high-level inferencing is also the basis of its ability to recall analogous memory episodes. High-level inferencing is the use of knowledge and rules about the world to build new beliefs about what is true. To understand a text, a reader must often make multiple inferences to understand the motives of actors and to causally connect actions that are unrelated on the basis of surface semantics alone. Complicating the inference process is the fact that language is often both lexically and conceptually ambiguous. A sentence that serves as a good example of many of the problems of high-level inferencing is the following:

John put the pot inside the dishwasher because the police were coming. **(Hiding Pot)**

Contrast this with the **Dinner Party** example mentioned earlier (*John put the pot inside the dishwasher because company was coming*). In **Dinner Party**, most people would infer that John transferred a Cooking-Pot inside a dishwasher to get the Cooking-Pot clean. In **Hiding Pot**, however, it seems more likely that John was trying to hide his Marijuana from the police. In this case, there are conflicts in the interpretation suggested by the first clause by itself (that John was cleaning a cooking-pot) and the final interpretation suggested by the first clause combined with the second clause (that John was hiding marijuana). This reinterpretation requires inferences like those shown in Table 1 to understand the most probable causal relationship between the actions in **Hiding Pot**.

To understand episodes such as **Dinner Party** and **Hiding Pot**, a system must be able to dynamically make chains of inferences and temporarily maintain them with a variable-binding mechanism. For example, a system must know about the general concept (or frame) of an actor transferring himself to a location (*coming*). To initially represent the phrase *police were coming* in **Hiding Pot**, the system must be able to temporarily maintain a particular instantiation of this Transfer-Self frame in which the Actor role (a variable) is bound to Police and the Location role is bound to some unknown location (which should later be inferred to be the location of John). The system must also have the general knowledge that when an actor transfers himself to a location, he ends up in the proximity of that location, which might be represented as the rule:

```
R1:   (Actor X Transfer-Self Location Y)
  == results-in ==>   (Actor X Proximity-Of Object Y)
```

Applying this rule to the instantiation of the police Transfer-Self would allow the system to make inference I5 in Table1, that the police will be in the proximity of John and his marijuana. Another rule the system must have to understand **Hiding Pot** is that an actor must be in the proximity of an object in order to see it:

```
R2:   (Actor X Proximity-Of Object Y)
  == precondition-for ==> (Actor X See-Object Object Y)
```

If rule R2 is applied to the new knowledge that the Police will be in the proximity of John, then the system infers that there is the potential for the Police to see John and his marijuana (I6). The rest of the inferences in Table1 required to understand **Hiding Pot** are the result of the application of similar knowledge rules about the world.

Even the ability to maintain variable bindings and apply general knowledge rules of the above sort is often insufficient for language understanding and other high-level cognitive tasks. This is because language is often ambiguous, as **Hiding Pot** illustrates, with several possible interpretations that must be discriminated. One of the fundamental problems in high-level inferencing is thus that of *frame selection* (Lytinen, 1984; Lange & Dyer, 1989). When should a system make inferences from a given frame instantiation? And when conflicting rules apply to a given frame instantiation, which should be selected? Only a system that can handle these problems will be able to address the following critical subparts of the frame selection problem:

Word-Sense Disambiguation: Choosing the contextually-appropriate meaning of a word. In **Dinner Party**, the word *pot* refers to a Cooking-Pot, but when **Hiding Pot** is presented, the evidence is that the interpretation should change to Marijuana.

Inferencing: Applying causal knowledge to understand the results of actions and the motives of actors. There is nothing in **Hiding Pot** that explicitly states that the police might see the pot (I6), or even that the police will be in proximity of it and John (I5). Nor is it explicitly stated what the police will do if they see he possesses Marijuana (I1, I2). Each of these assumptions must be inferred from the facts specified in the text.

Concept Refinement: Instantiating a more appropriate specific frame from a general one. In **Dinner Party**, the fact that the pot was put inside a dishwasher tells us more than the simple knowledge that it was put inside a container. In contrast, the salient point in **Hiding Pot** is that it is inside of an opaque object (I8), which allows us to infer that the police will not be able to see it (I9).

Plan/Goal Analysis: Recognizing the plan an actor is using to fulfill his goals. In **Dinner Party**, John has put the pot into the dishwasher as part of the $Dishwasher-Cleaning script (a stereotypical sequences of actions) to satisfy his goal of getting the pot clean, perhaps itself serving as part of his plan to prepare for company coming over. In **Hiding Pot**, however, it appears that John has put the pot into the dishwasher to satisfy his sub-goal of hiding the pot from the police (I4), which is part of his overall goal of avoiding arrest (I3).

High-level inferencing is complicated by the effect of additional context, which often causes a *reinterpretation* to competing frames. For example, the interpretation of **Hiding Pot** can change again if the next sentence is:

P3: They were coming over for dinner in half an hour.

P3 provides more evidence for the possibility that John was trying to clean the pot to prepare for dinner, perhaps causing the word *pot* to be reinterpreted back to Cooking-Pot, as in **Dinner Party**. These examples clearly point out two sub-problems of frame selection, those of *frame commitment* and *reinterpretation.* When should a system commit to one interpretation over another? And if it does commit to one interpretation, how does new context cause that interpretation to change?

The issues of word-sense disambiguation and inferencing have been relatively well explored in psychological experiments. Experiments have shown that many potential meanings of a word (e.g., Cooking-Pot, Marijuana, and Planting-Pot for the word *pot*) are primed immediately after the word is read — causing subjects, for example, to respond more quickly to words closely related to each of the meanings than to non-primed words (Swinney, 1979; Till, Mross, & Kintsch, 1988). However, after no more than a second, the contextually-appropriate meaning of the word becomes significantly more primed than the non-appropriate meanings. The meaning of the ambiguous word becomes constrained by the lexical environment in which it appears, which is crucial to the selection of its contextually appropriate sense (Glucksberg, Kreuz, & Rho, 1986; Till, Mross, & Kintsch, 1988).

Semantic reinterpretation is necessary when an old lexical interpretation is no longer appropriate to a new linguistic context. While there have been very few studies of this phenomenon, it would seem reasonable that previously activated word meanings would not immediately decay to baseline activation. This residual activation might play a role in the ability to reinterpret word meanings when new linguistic contexts are encountered, as in **Hiding Pot**.

2. Related Models of Comprehension and Memory Retrieval

In REMIND, the understanding mechanism constructs an elaborated interpretation of its input that not only serves as the model's representation of the meaning of the text, but is also used as a cue for episodic memory retrieval. The language-understanding component of the system must be able to (a) perform the high-level inferencing necessary to create a causal plan/goal analysis of the cue, (b) dynamically store the complex structured cue representation, and (c) perform lexical disambiguation (and possible reinterpretation) to select the most contextually-appropriate representation. Thus, the brunt of the work in an integrated language understanding and memory retrieval system falls upon the language understanding part of the model. In this section we discuss several related symbolic and connectionist approaches to these language understanding problems and give a brief overview of previous models of memory retrieval.

2.1 Symbolic Rule-Based Systems

Symbolic rule-based systems have been the most successful AI models at performing the high-level inferencing necessary for natural language understanding. A good example is BORIS (Dyer, 1983), a program for modeling in-depth understanding of relatively long and complex stories. BORIS has a symbolic knowledge base containing knowledge structures representing various actions, plans, goals, emotional affects, and methods for avoiding planning failures. When a story is read in, BORIS fires rules from its knowledge base to infer additional story information. This allows BORIS to form an elaborated representation of the story, about which it can 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 (cf. Schank & Abelson, 1977; Lebowitz, 1980; Wilensky, 1983; Lytinen, 1984; Reeves, 1991).

While traditional symbolic models have demonstrated an ability to understand relatively complex stories (albeit in limited domains), these models encounter difficulty when trying to resolve and reinterpret ambiguous input. One solution has been to use expectation-based conceptual analyzers, as in such models as CA (Riesbeck, 1975) and BORIS (Dyer, 1983). These systems use bottom-up or top-down *requests* or *demons* that are activated as words are read in. A word is disambiguated when one of the request rules fires. An example of a bottom-up request that might be used to disambiguate the word *pot* would be:

```
If the context involves Cleaning
   then interpret pot as a Cooking-Pot.
```

Once such a request is fired, the interpretation chosen is generally used throughout the rest of the inferencing process, and the word is thrown away. However, this makes it impossible to reinterpret the word if the context changes, such as in **Hiding Pot**. A partial answer might be to keep words around in case a new context causes another disambiguation request to fire. However, this solution creates a different problem — how to decide between conflicting disambiguation rules. For example, one cannot simply specify that the *pot* disambiguation request involving the Police context always has a higher priority than the request involving the Cleaning context, because police can be in the same place as cooking pots (e.g., if **Hiding Pot** was followed by *They were coming over for dinner in half an hour.*) As the amount of knowledge stored in the system grows, the number of disambiguation requests needed grows with them, producing even more conflicts. Moreover, because rule application in traditional symbolic models is fundamentally serial, these systems dramatically slow down as the number of inferencing and disambiguation rules increases.

Partially because they avoid such problems, connectionist networks have significant potential advantages over traditional symbolic approaches to language understanding. Their conceptual knowledge is stored entirely in an interconnected network of units whose states are computed in parallel. The activation of these units is calculated solely by local update functions which are based on their previous state and the other units to which they are connected. As a result, a major portion of the understanding process is potentially controlled by a relatively simple, local spreading-activation mechanism, instead of by a large collection of brittle and possibly ad hoc rules.

2.2 Marker-Passing Networks

Marker-passing models operate by spreading symbolic markers in parallel across labeled semantic networks in which concepts are represented by individual nodes. Possible interpretations of the input are formed when marker propagation results in a path of units connecting words and concepts from the input text. Like rule-based systems, marker-passing systems are able to perform much of the high-level inferencing necessary for language understanding because of the symbolic information held in their markers and networks (cf. Charniak, 1986; Riesbeck & Martin, 1986; Granger, Eiselt, & Holbrook, 1986; Norvig, 1989; Kitano, Tomabechi, & Levin, 1989). The primary advantage of marker-passing networks over traditional symbolic, rule-based systems is that their massively-parallel marker-passing process allows them to generate all of the different possible interpretations of a text in parallel. This is a tremendous advantage for ambiguous texts such as **Hiding Pot** and for more complex stories.

Marker-passing systems have many of the same problems as traditional symbolic systems in performing disambiguation and reinterpretation. Because of the generally all-or-none symbolic nature of the inference paths generated by the marker-passing process, these systems have problems choosing the most contextually-sensible interpretation out of all the paths that they generate. Most marker-passing models attempt to deal with this problem by using a separate symbolic path evaluation mechanism to select the best interpretation. Unfortunately, the marker-passing process generally creates an extremely large number of *spurious* (i.e., unimportant or logically impossible) inference paths, which often represent over 90% of the paths generated even for small networks (Charniak, 1986). As network size increases to include more world knowledge, there is a corresponding explosion in the number of paths generated. Because path evaluation mechanisms work serially, marker-passing systems' advantage of generating inference paths in parallel is substantially diminished. This explosion of generated connections and the generally all-or-none nature of marker-passing inference paths become especially difficult problems when applying marker-passing systems to ambiguous natural language texts (Lange, 1992)[2].

2.3 Distributed Connectionist Networks

Distributed connectionist (or PDP) models represent knowledge as patterns of activation within massively parallel networks of simple processing elements. Distributed connectionist models have many desirable properties, such as learning rules that allow stochastic category generalization, noise-resistant associative retrieval, and robustness against damage (cf. Rumelhart, Hinton, & McClelland, 1986).

A good example of how distributed connectionist models have been used to model language understanding is provided by the case-role assignment model of McClelland and Kawamoto (1986). The main task of their model is to learn to assign 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 Boy in 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 Rock in the Instrument role. Their network is also trained to perform lexical disambiguation, for example, 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 new inputs that are similar to the training sentences.

One of the main limitations of McClelland and Kawamato's model for language understanding is that it can only successfully analyze direct, one-step mappings from the input to the output. This limits the model to sentences that can be understood and disambiguated based solely upon the surface semantics of the input. Two distributed connectionist models that get around this limitation are those of Miikkulainen and Dyer (1991) and St. John (in press). 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 texts based on stereotypical scripts and script-like stories (Schank & Abelson, 1977). Both models have the lexical disambiguation abilities of McClelland and Kawamoto's model, but are also 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 distributed connectionist models to handle more complex language. Both the Miikkulainen/Dyer and the St. John model work by resolving constraints from the context of the input to recognize one of their trained scripts and to 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* — producing chains of inferences over simple known rules, with each inference resulting in a

2Partial solutions to these problems have been proposed by several researchers using *hybrid* marker-passing networks that include some aspects of spreading-activation (cf. Kitano *et al.*, 1989; Hendler, 1989).

potentially novel intermediate state (Touretzky, 1990). Most importantly, the problem of ambiguity and the exponential number of potential causal connections between two or more events requires that multiple paths be explored in parallel (the forte of marker-passing networks). It remains to be seen whether a single blended activation pattern across the bank of hidden units in a recurrent network can solve this problem by simultaneously holding and making dynamic inferences for multiple, never-before encountered interpretation chains.

Other distributed models explicitly encode variables and rules, such as the models of Touretzky and Hinton (1988) and Dolan and Smolensky (1989). Consequently, such *rule-implementing* distributed models are able to perform some of the dynamic inferencing necessary for language understanding. However, the types of rules they can currently encode are generally limited. More importantly, like traditional rule-based systems, they are *serial at the knowledge level* — i.e., they can fire only one rule at a time. As previously mentioned, this is a serious drawback for natural language understanding, particularly for ambiguous text, in which the often large number of multiple alternative inference paths must be explored in parallel (Lange, 1992).

2.4 Structured Spreading-Activation Models

Structured (or localist) spreading-activation models are connectionist models that represent knowledge in semantic networks like those of marker-passing networks, but in which the nodes are simple numeric units with weighted interconnections. The activation on each conceptual node generally represents the amount of *evidence* available for its concept in a given context. As in marker-passing networks, structured connectionist networks have the potential to pursue multiple candidate interpretations of a story in parallel (i.e. be parallel at the knowledge level) as each interpretation is represented by activation in different local areas of the network. Unlike pure marker-passing networks, however, the evidential nature of structured spreading-activation networks make them ideally suited to perform lexical disambiguation. Disambiguation is achieved automatically as related concepts under consideration provide graded activation evidence and feedback to one another in a form of constraint relaxation (cf. Cottrell & Small, 1982; Waltz & Pollack, 1985; Kintsch, 1988).

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

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

The word *star* could easily be 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 structured network Waltz and 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 Astronomer to Celestial-Body.

Until recently, the applicability of structured connectionist models to natural language understanding has been severely hampered because of their difficulties representing dynamic role-bindings and performing inferencing. The basic problem is that the evidential activation on structured networks' conceptual units gives no clue as to *where* that evidence came from. 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 bindings to make the chains of inferences necessary for understanding more complex texts. Thus, unlike marker-passing systems, most structured connectionist models have been limited to simple language processing tasks that can be resolved solely on the surface semantics of the input.

A way of compensating for the lack of dynamic inferencing abilities in spreading-activation networks is to use a symbolic processing mechanism external to the spreading-activation networks themselves to perform the variable binding and inferencing necessary for language understanding. Such a spreading-activation/symbolic hybrid has been used in Kintsch's (1988) construction-integration model of language comprehension. This system uses a traditional symbolic production system to build symbolic representations of the alternative interpretations of a text.

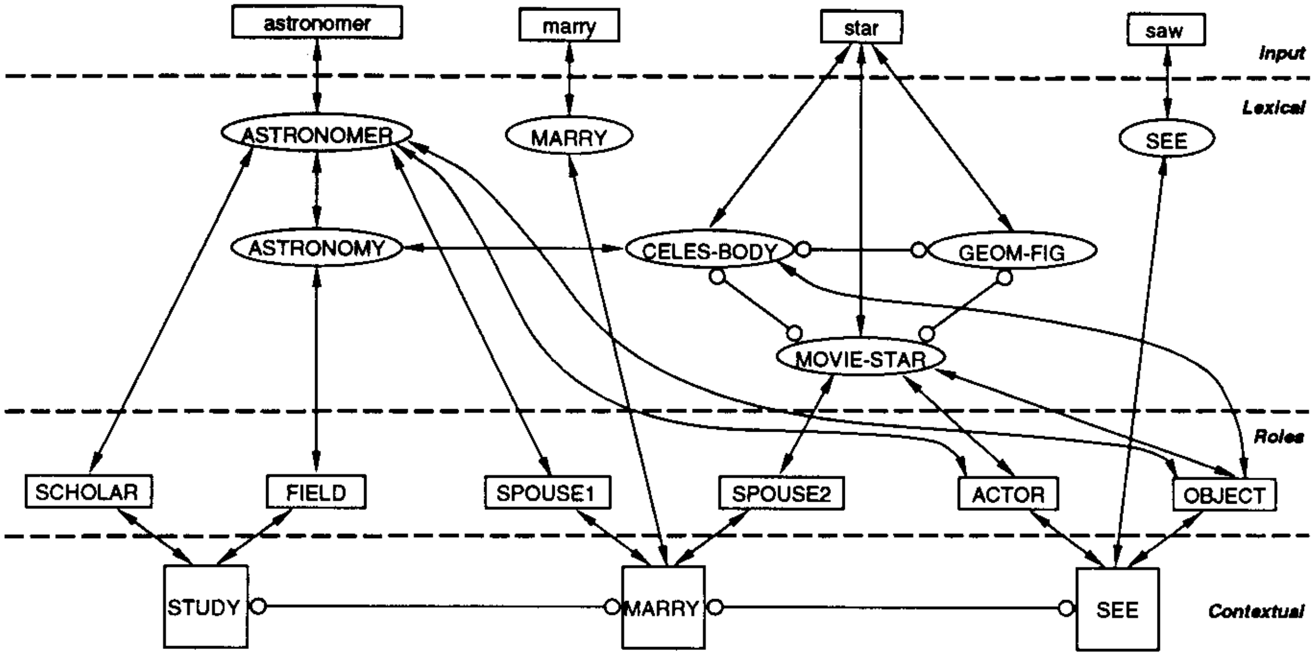

Figure 1. Localist spreading-activation network based on Waltz and Pollack (1985). Lines with arrows are excitatory connections; lines ending with open circles are inhibitory.

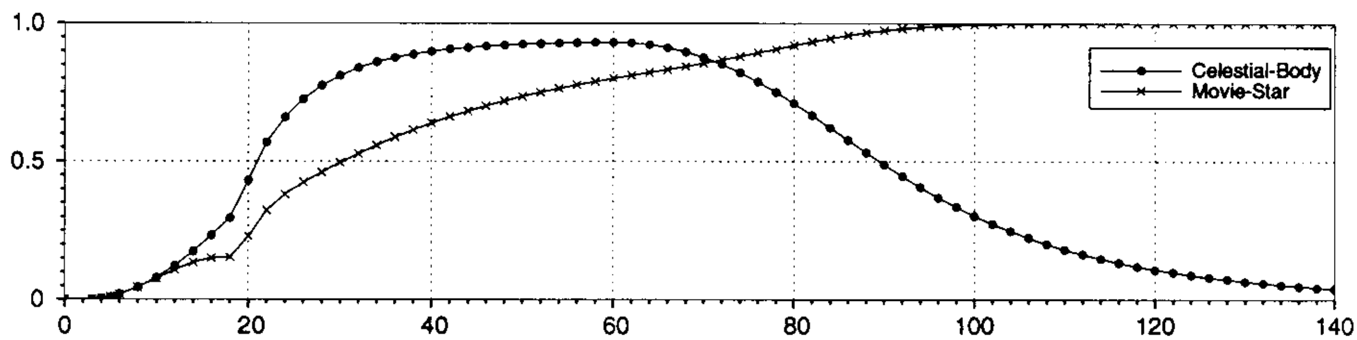

Figure 2. Activations of meaning of word *star* after *astronomer married star* is clamped for network in Figure 1.

These representations are then used to construct a spreading-activation network in which the different interpretations compete to integrate contextual constraints. The integration of constraints with spreading-activation in the network allow Kintsch's model to correctly disambiguate and interpret input sentences. A somewhat similar approach is taken by ACT* (Anderson, 1983), a psychologically-based spreading-activation model of language understanding, fact encoding and retrieval, and procedure encoding and retrieval. Kintsch's and Anderson's models both illustrate many of the impressive emergent properties of spreading-activation networks for modeling realistic language understanding, such as their ability to model the time course of lexical disambiguation in a way consistent with psychological evidence (e.g., Swinney, 1979). However, if a mechanism internal to the networks (instead of an external symbolic production system) could be found to construct text inferences, the parsimony and psychological realism of structured spreading-activation networks would be greatly increased.

Recently, a number of researchers have shown how structured connectionist models can handle some variable binding and inferencing abilities within the networks themselves (e.g., Barnden, 1990; Holldobler, 1990; Shastri & Ajjanagadde, in press; Sun, in press). Most of these models, however, have not been applied to language understanding and have no mechanisms for handling ambiguity. An exception is ROBIN (Lange & Dyer, 1989), a structured spreading-activation model that propagates *signatures* (activation patterns that identify the concept bound to a role) in order to generate all possible interpretations of an input text in parallel. At the same time, ROBIN uses the network's evidential constraint satisfaction to perform lexical disambiguation and selection of the contextually most plausible interpretation. Thus, ROBIN is able to perform high-level inferencing and disambiguation within the structure of a single network, without the need for external symbolic processing. Given these abilities, such a structured spreading-activation model seems a promising place to start for building an integrated language understanding and memory retrieval model.

2.5 Memory Retrieval Models

The process of memory retrieval has generally been explored in isolation from the process of language understanding. Storage and retrieval of complex episodes requires many of the same abilities to represent and handle structural relationships and variable bindings that natural language understanding does. Because connectionist models have had difficulties handling complex structural relationships in general, few attempts have been made to build connectionist retrieval models for the type of high-level episodes discussed in this paper. Nonetheless, a few models have shown the potential value of connectionist models for memory storage and retrieval. For example, COPYCAT (Hofstadter & Mitchell, in press) uses connectionist constraint-satisfaction in solving letter-string analogy problems. Although the retrieval portion of COPYCAT only retrieves simple concepts and not memory episodes, it seems to exhibit some of the fluidity of concepts and perception apparent in human analogical reasoning. Miikkulainen (in press) shows how a variant of distributed connectionist topological feature maps (Kohonen, 1984) can be used to store and retrieve the script-based stories that it has understood using recurrent distributed networks. Besides showing how purely-distributed connectionist models can store and retrieve multiple-sentence episodes, Miikkulainen's model exhibits a number of features of human episodic memory, such as certain kinds of memory confusions and recency effects. Although connectionist models such as COPYCAT and Miikkulainen's DISCERN are currently able to store only relatively simple or stereotypical episodes, they do illustrate their promise for psychologically-plausible memory retrieval.

As for natural language understanding, symbolic models have had the greatest success in modeling retrieval of complex, high-level memory episodes. Case-based reasoning (CBR) models (cf. Hammond, 1989; Riesbeck & Schank, 1989) form the largest class of symbolic memory retrieval models. In CBR models, memory access is performed by recognition of meaningful *index patterns* in the input that allow retrieval of the episodes (or cases) most likely to help them solve their current problem. An analysis phase is usually performed to determine the indices that are most important for finding relevant cases for a particular problem, such as cases that share similar plans, goals, enabling preconditions, or explanation failures. In addition, CBR models are usually careful to retrieve *only* those cases that will help find a solution, explicitly rejecting cases that do not. CBR models are therefore generally models of expert reasoning within a given domain of expertise, rather than models of general human reminding. It is quite possible that expert memory retrieval may be satisfactorily modeled by such methods. However, general reminding seems to be substantially messier, being affected by not only by the sort of useful abstract indices used in CBR models, but also by superficial semantic similarities that often lead to quite *inexpert* remindings. Further, the problem of selecting and recognizing appropriate indices becomes substantially more difficult when reading ambiguous texts outside of limited expert domains.

General, non-expert reminding has been modeled in systems such as ARCS (Thagard, Holyoak, Nelson, & Gochfeld, 1990) and MAC/FAC (Gentner & Forbus, 1991). These systems model retrieval without using specific indexing methods. Instead they retrieve episodes whose representations share superficial semantic similarities with retrieval cues, with varying degrees of preference towards retrieval of episodes that are also analogically similar or structurally consistent. However, unlike most CBR models, these systems do not specify how they construct the representation of retrieval cues from a source input or text, and so cannot explain how inferences and comprehension affect reminding.

Theoretically, REMIND lies somewhere between case-based reasoning models and general analogical retrieval models such as ARCS and MAC/FAC. Like ARCS and MAC/FAC, REMIND is meant to be a psychologically-plausible model of general human reminding, and therefore takes into account the prevalence of superficial feature similarities in remindings. However, we believe that many of the types of high-level planning and thematic knowledge structures used as indices in case-based reasoning systems also have an important effect on reminding. REMIND is thus partially an attempt to bridge the gap between case-based and analogical retrieval models. As it turns out, this gap is naturally bridged when the same spreading-activation mechanism is used to both understand cues and to retrieve episodes from memory. Using the same mechanism for both processes causes retrieval to be affected by all levels that a text was understood with, as hypothesized by Schank (1982). This is the case in REMIND, in

Table 2. Simplified definition of the frame representing the state Inside-Of.

```
(FRAME Inside-Of
  State
  :Roles   (Object    (Physical-Object  0.05))
           (Location  (Container-Object 0.30))
           (Planner   (Human            0.05))
  :Phrase       (<S_is inside of_DO>    1.0 (Object Subject)  (Location Direct-Object))
  :Result-Of    (Transfer-Inside        1.0 (Object Object)   (Location Location) (Planner Actor))
  :Refinements  (Inside-Of-Dishwasher   1.0 (Object Object)   (Location Location) (Planner Planner))
                (Inside-Of-Opaque       1.0 (Object Object)   (Location Location) (Planner Planner))
                (Inside-Of-Carwash      1.0 (Object Object)   (Location Location) (Planner Planner)))
```

which the understanding mechanism is given the superficial features and actions of a text and attempts to explain them by inferring the plans and goals being used — causing long-term memory episodes to be activated by both.

3. Language Understanding in REMIND

REMIND is a structured spreading-activation model that integrates language understanding and memory retrieval. It is an extension of ROBIN (Lange & Dyer, 1989; Lange, 1992), a structured connectionist model that performs high-level inferencing and disambiguation to build interpretations of syntactically-parsed input for short texts such as **Hiding Pot** and **Dirty Magazine**. These interpretations are then added to the network to encode the models long-term memory episodes.

In REMIND, memory retrieval is a natural side-effect of using spreading-activation to perform language understanding. The knowledge structures used to understand an input cue activate similar episodes that were understood and stored in the network earlier. For example, **Dirty Magazine** becomes active when **Hiding Pot** is being understood because both involve hiding to avoid punishment. An episode is retrieved from memory when there are enough similarities between it and a cue's interpretation to cause it to become the most highly-active episode in the network. Because both inferencing and memory retrieval occur within a single spreading-activation network, these processes strongly interact and affect each other, as appears to be the case in human memory. In this section, we give an overview of how REMIND performs high-level text inferencing and disambiguation. A more detailed description is provided in Lange and Dyer (1989).

3.1 Knowledge Given To REMIND

REMIND, like ROBIN, uses structured networks of simple connectionist units to 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. General knowledge rules used for inferencing are encoded as interconnected pathways between corresponding roles. The knowledge base of frames and rules consists of the causal dependencies relating actions, plans, goals, and scripts (Schank & Abelson, 1977) necessary for understanding stories in a limited domain. The knowledge base is hand-built, as in most structured connectionist models. However, there is no information in the knowledge base about specific episodes (such as **Hiding Pot**, **Dinner Party**, and **Dirty Magazine**) that the networks will be used to understand.

Table 2 gives an example of how knowledge is defined in REMIND. It defines the conceptual frame Inside-Of, which represents the general state of one object being inside of another. Inside-Of has three roles: an Object that is inside of something (which must be a Physical-Object), a Location that the Object is inside of (which must be a Container-Object), and a Planner that may have caused the state to be reached (which must be a Human). The rest of Table 2 defines Inside-Of's *relations* to other frames. The knowledge represented here is that it is (a) directly accessed by the phrase <S_is inside of_DO> (as in *The fork is inside of the dishwasher)*, (b) a Result-Of the action Transfer-Inside, and (c) has several possible concept refinement frames: Inside-Of-Dishwasher, Inside-Of-Opaque and Inside-Of-Carwash.

Refinements (short for concept refinements, an inverse of the *is-a* relation) of frames are useful because they allow more specific inferences to be made when role-bindings are known (Lytinen, 1984). For example, if the network has inferred that a cooking utensil is inside of a dishwasher (Inside-Of-Dishwasher), a likely inference is that it is about to cleaned. If the network has inferred that any object is inside of an opaque object (Inside-Of-Opaque), the network can infer that the object is blocked from sight.When multiple frames are defined as alternatives for a given relation to a frame, as in the multiple refinements of Inside-Of, they are defined as *mutually exclusive* relations which compete for selection as the relations instantiation at any given time. For example, although there are multiple possible plans-for the goal of Satisfy-Hunger (e.g., $Restaurant, $Eat-At-Home, etc.), generally only one will be used as the plan for a *given* instance of somebody wanting to satisfy his hunger in a particular story.

The relations and their role correspondences shown in Table 2 also define the network's general knowledge rules, such as the following:

```
R3: (Subject X <S_is inside of_DO> Direct-Obj Y)
  == phrase ==>
 (Object X Inside-Of Location Y)
```
(The phrase X is inside of Y means that object X is inside of object Y).

```
R4: (Actor X Transfer-Inside Object Y Location Z)
  == results-in ==>
 (Object Y Inside-Of Location Z Planner X)
```
(When an actor X transfers an object Y into a location Z, then object Y is inside of location Z).

Finally, the numbers in Table 2 represent the connection weights (ranging from 0 to 1) from each of the related concepts to Inside-Of, and are chosen on the basis of how much evidence they provide. For example, if an object has just been transferred inside of something else (Transfer-Inside), then the network can definitely infer that the object is Inside-Of it. Therefore, the weight from Transfer-Inside to Inside-Of is maximal (1.0). If something that is a container (Container-Obj) has been mentioned in a story, then there is some, though not certain, evidence that something is inside of it, so a corresponding middling weight of 0.3 from Container-Obj to Inside-Of's Location role is given. On the other hand, a very small weight (0.05) is given from Physical-Object to Inside-Of's Object role, since mere mention of any particular physical object does not very strongly imply Inside-Of. The actual weights chosen are clearly arbitrary. What is important is that they be in a *range* reflecting the amount of evidence the concepts provide for their related concepts in a certain knowledge base.

3.2 Structure of REMIND

The knowledge given to REMIND is used to *construct* the network before any processing begins. As with other structured connectionist models, a single node in the network represents each frame or role. Relations between concepts are represented by weighted connections between the nodes. Activation on frame and role nodes is *evidential*, corresponding to the amount of evidence available from the current context for that concept. However, as described earlier, simply representing the amount of evidence available for a concept is not sufficient for complex inferencing tasks. Solving the variable binding problem requires a way to *identify* the concept that is dynamically bound to a role. Furthermore, the network's structure must allow such role-bindings to propagate across the network to dynamically instantiate inference paths and form an elaborated representation of the input.

3.3 Variable Binding With Signatures

Representation of variables and role-bindings is performed in REMIND by network structure that processes *signatures* — activation patterns that uniquely identify the concept bound to a role (Lange & Dyer, 1989). 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.

An example of signatures is shown in Figure 3, which shows the concept nodes for the concepts Police, John, and Dishwasher (on the lower plane) and their associated signature units (banks of units on the top plane). Here signatures are shown as unique six-unit distributed patterns, with different levels of activation being represented by different levels of gray. The figure also shows some of the units for the frame Transfer-Inside and their activation values when its Actor is bound to John. The *virtual binding* of Transfer-Inside's Actor role to John is represented by the fact that its binding units have the same activation pattern as John's signature. The binding banks for the

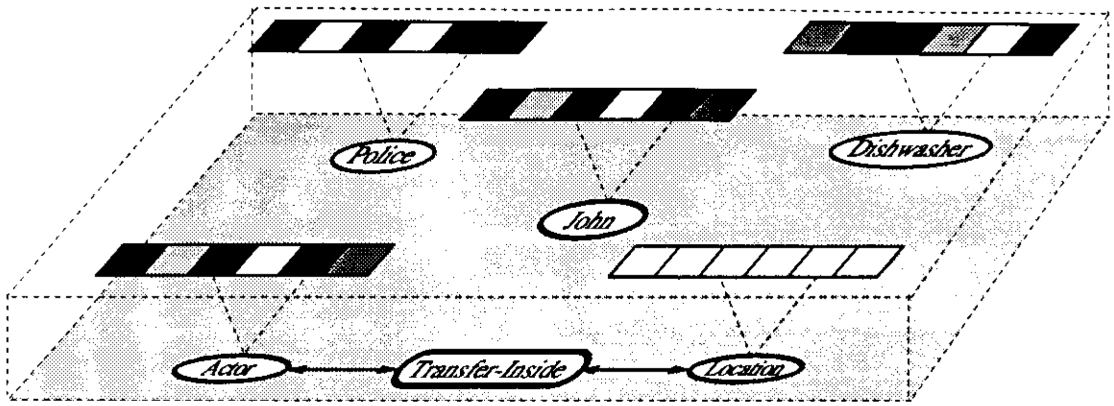

Figure 3. Examples of signature patterns (banks of units on top plane) for concepts (ovals on lower plan). Actor and Location roles and of the Transfer-Inside frame and their binding units are also shown.

Location role have no activation because this role is currently unbound. 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. The same binding units could, at another time, hold a different virtual binding, simply by having the activation pattern of another concept's signature.

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 reduced semantic representations of the concept for which they stand (Figure 3). Having the signatures represented as distributed activation patterns carrying semantic information may allow their future use as inputs for local distributed learning mechanisms after they have been propagated for inferencing (Lange & Dyer, 1989). For simplicity, however, REMIND's simulations are currently run with signatures simply being unique arbitrarily-generated scalar values (e.g. 6.8 for Marijuana and 9.2 for Cooking-Pot).

3.4 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. Figure 4 shows how the network's structure accomplishes this and automatically propagates signatures to fire rules (such as R4). Evidential activation for disambiguation is spread through the paths between conceptual units on the bottom plane, e.g., Transfer-Inside and its Object role. Signature activation for dynamic role-bindings is spread across the parallel paths of corresponding binding units (solid black circles) on the top plane. For simplicity, the signatures in the figure are uniquely-identifying scalar values. Units and connections for the Actor, Planner, and Location roles are not shown. As shown here, there are actually multiple binding units per role to allow simultaneous propagation of ambiguous bindings, such as the multiple meanings of the word *pot*. 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 a phrase such as *John put the pot inside the dishwasher* (**P1**) is presented, the lexical concept nodes for each of the words in the phrase are clamped to a high level of evidential activation. This directly provides activation for the 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. For example, the binding units of Transfer-Inside's Object are clamped to the signature activations (6.8 and 9.2) of Marijuana and Cooking-Pot, representing the candidate bindings from the word *pot* (Figure 4)[3]. An alternative input, such as *George put the cake inside the oven,* would be represented by clamping the signatures of its bindings (i.e., George, Cake, and Oven) instead. A completely different set of inferences would then ensue.

---

[3]REMIND and ROBIN do not currently address the problem of deciding upon the original syntactic bindings, that is that *pot* is bound to the Object role of phrase **P1**. Rather, their networks are initially given these bindings and then use them for high-level inferencing.

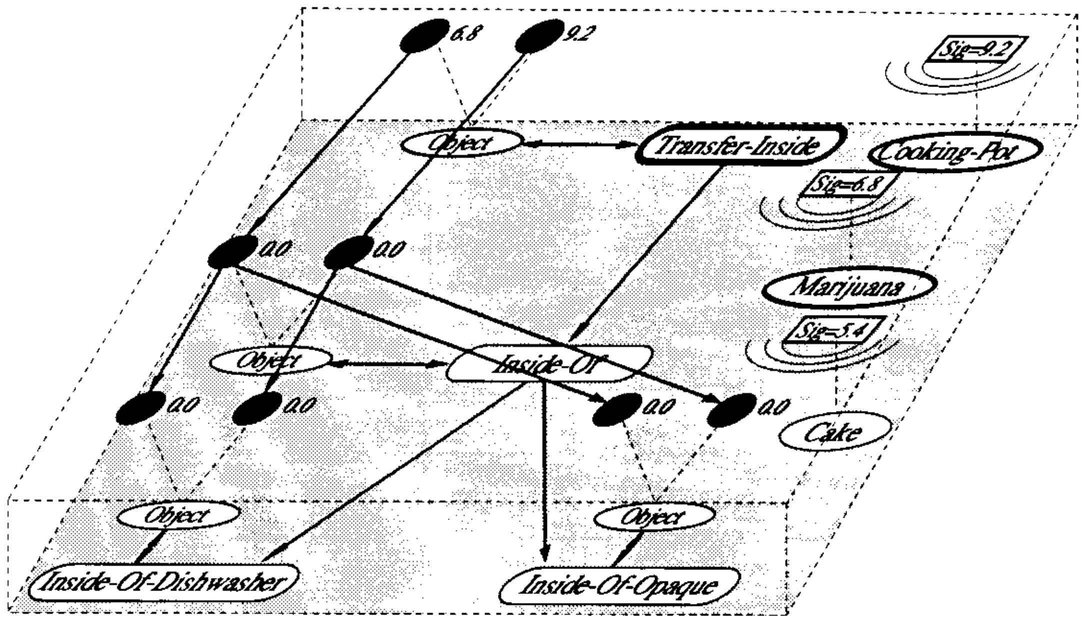

Figure 4. Simplified ROBIN/REMIND network segment showing parallel paths over which evidential activation (bottom plane) and signature activation (top plane) are spread for making inferences. The figure shows the initial activation and clamping for the first phrase of **Hiding Pot** (*John put the pot inside the dishwasher*). Signature nodes (outlined rectangles) and binding nodes (solid black circles) are in the top plane. Thickness of conceptual node boundaries (ovals) in the bottom plane represents their levels of evidential activation. Node names do not affect the spread of activation in any way. They are used only to initially set up the network's structure and to aid in analysis.

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 structured networks. However, the activation of the binding units is equal to the maximum of their unit-weighted inputs so that signatures can 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, because gated connections (not shown) ensure that two different signatures do not reach the same binding node.

As activation starts to spread after the initial clamped activation values in Figure 4, 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 5), because 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.

By propagating signature activations from the binding nodes of Transfer-Inside to the binding nodes of Inside-Of, the network has made its first inference. Because of the signatures now on Inside-Of's binding nodes, the network not only represents that something is inside of something else, but also represents exactly which thing is inside the other. REMIND continues to make subsequent inferences from the activations of this new knowledge. Evidential and signature activation spreads, in parallel, from Inside-Of to its refinements Inside-Of-Dishwasher and Inside-Of-Opaque and their corresponding binding units (see Figure 5), on through the rest of the network[4]. Figure 6 shows an overview of the signature bindings in a portion of the network after presentation of the 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 1, with most being shown in the figure.

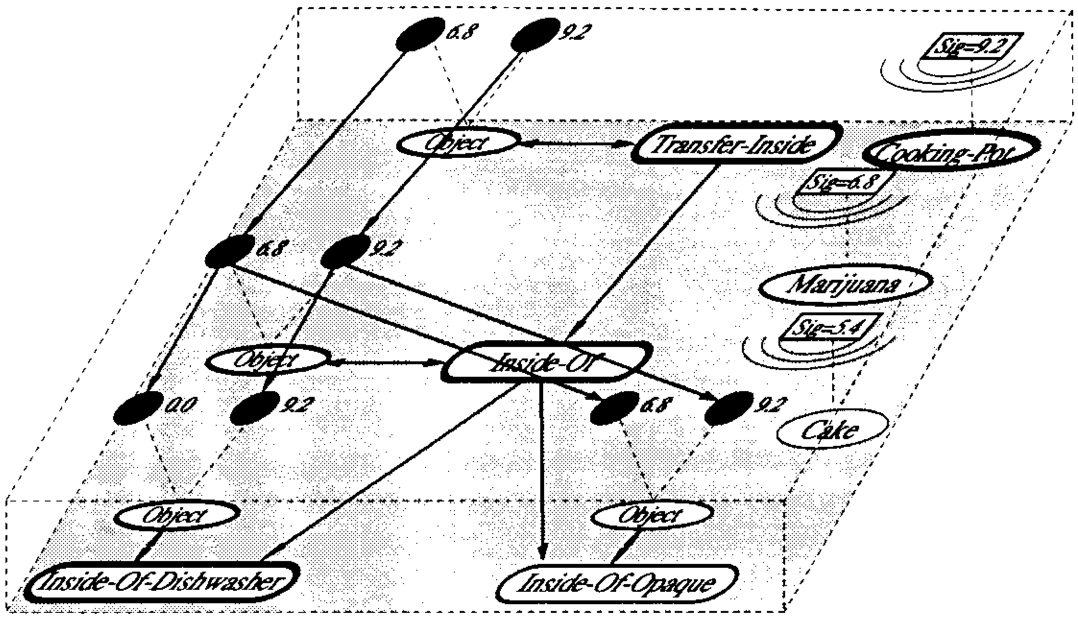

Figure 5. Simplified ROBIN/REMIND network segment showing activation midway through processing **Hiding Pot**. At this time, Cooking-Pot and Inside-Of-Dishwasher have higher evidential activations than Marijuana and Inside-Of-Opaque, as is illustrated by their thicker ovals.

3.5 Disambiguation and Reinterpretation

REMIND's propagation of signature activations dynamically instantiates candidate inference paths in parallel in much the same way as marker-passing systems and the structured connectionist binding mechanisms of Shastri and Ajjanagadde (in press) and Sun (in press). However, as described earlier, natural language understanding requires more than simple basic variable binding and rule-firing capabilities — it also requires the ability to resolve ambiguities and select between the large number of candidate inference paths instantiated by rule-firing. This is handled in REMIND by the evidential activation that spreads in parallel with signature bindings.

If this were a marker-passing system constructing an internal representation of **Hiding Pot**, 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. At the end of processing, the path evaluator would also have to recognize that Marijuana should be selected over the Cooking-Pot and Planting-Pot bindings throughout the network.

Such disambiguation is performed entirely within REMIND's network without resorting to a separate path-evaluation module. Instead, the evidential portion of the network (Figure 5) decides between the competing inference paths that have been instantiated by signature activation. The activations of the conceptual frame nodes are always approximately proportional to the amount of evidence available for them in the current context from their bindings and related frames. REMIND's interpretation of its input is the most highly-activated path of frame units and their bindings when the network settles[5].

Often there are multiple possible competing interpretations for a given frame. This occurs when there are multiple plans to achieve a goal or multiple refinements for a frame (e.g., the Inside-Of-Dishwasher and Inside-Of-Opaque refinements of Inside-Of). In these cases, the most highly activated interpretation that has been instantiated with compatible signature role-bindings is chosen as part of the inference path. Similarly, when there are mul-

---

[4]The reader may note that he signature for Marijuana did not reach the binding units of Inside-Of-Dishwasher in Figure 5. This is due to additional structure of gated links that encode knowledge about what kind of concepts can be bound to the roles of particular frames — such that only concepts that are refinements of Cooking-Utensils are prototypically cleaned as the Object in Inside-Of-Dishwasher. These *selectional restrictions* and their importance are described further in Section 3.7.

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

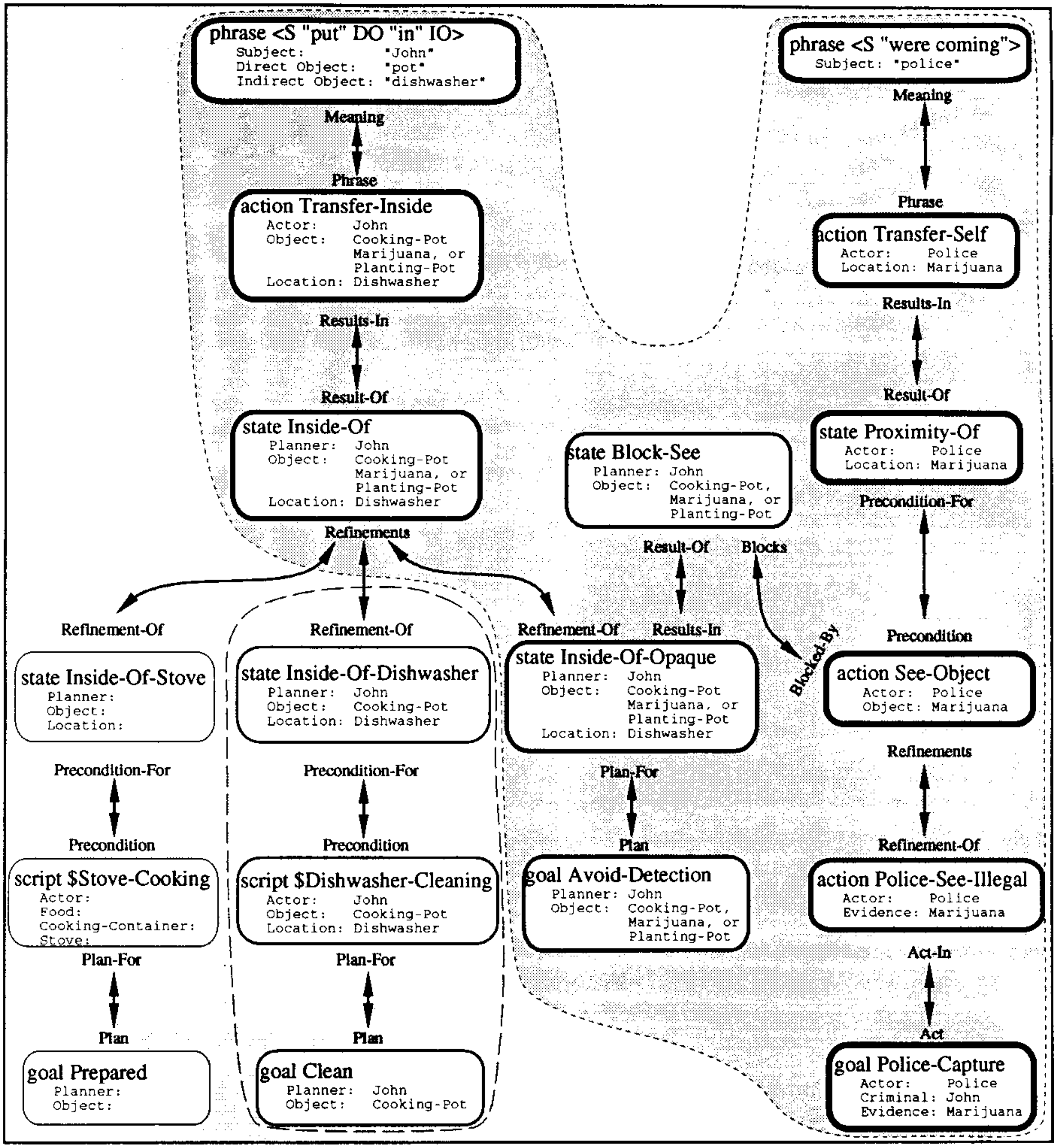

Figure 6. Overview of a small portion of a ROBIN/REMIND network showing inferences made after clamping of inputs for phrases P1 and P2 of **Hiding Pot**. Thickness of frame boundaries shows the amount of evidential activation on the frames conceptual nodes. Role fillers shown are the ones dynamically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

tiple possible bindings for a role, the binding chosen at any given time is the one whose concept has the highest level of evidential activation.

Figure 7 illustrates how evidential activation works through constraint satisfaction to disambiguate meanings and interpretations. The evidential activations of the competing meanings of *pot* and refinements of Inside-Of change

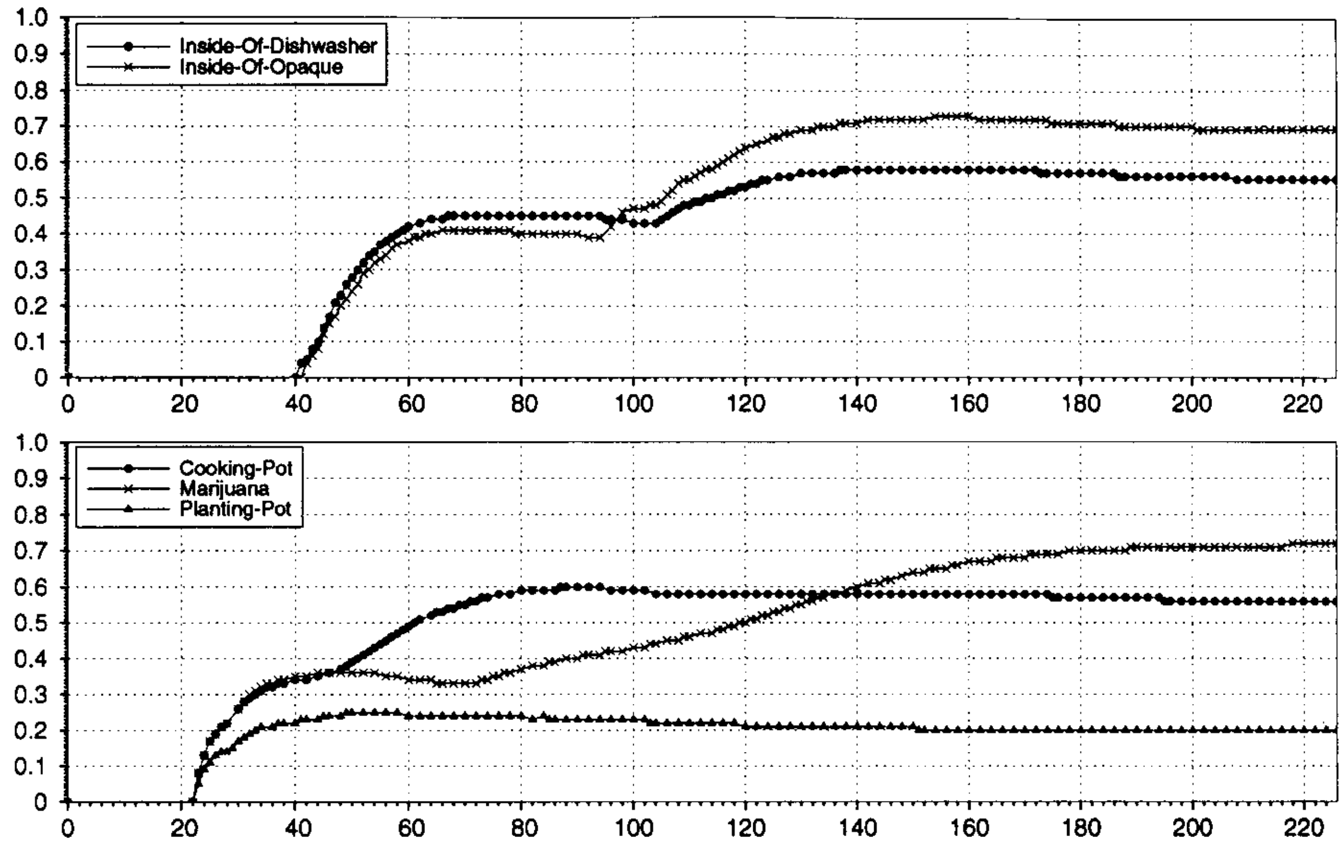

Figure 7. Evidential activations for meanings of *pot* and of competing refinements of Inside-Of after presentation of *John put the pot inside the dishwasher* (**P1**) at cycles 1 through 31 and *the police were coming* (**P2**) at cycles 51 through 61.

during the processing of **Hiding Pot**. Initially there is more evidence for the interpretation that John was trying to clean a cooking pot. This is shown by the fact that after Inside-Of-Dishwasher becomes activated at about cycle 60, Cooking-Pot becomes more highly activated than Marijuana or Planting-Pot. Input for the second phrase of **Hiding Pot** (*because the police were coming*) is presented at cycles 51 through 61. The evidential activation levels shown by the thickness of conceptual node boundaries in Figure 5 correspond to the activations at cycle 90. The inferences about the police propagate through Transfer-Self, Proximity-Of, See-Object, and Block-See, until they reach Inside-Of-Opaque (see Figure 6). This occurs at about cycle 95. By about cycle 160, reinforcement from the Block/See/Police/Capture path causes Inside-Of-Opaque to become more activated than Inside-Of-Dishwasher, and Marijuana to become more highly activated than Cooking-Pot. Thus, REMIND's interpretation of **Hiding Pot** is that John was trying to avoid detection of his Marijuana from the police by hiding it inside of an opaque dishwasher. The final inference path interpretation is shown in the darkly shaded area of Figure 6.

3.6 Activation Control

A major issue for all structured connectionist networks is controlling the spread of activation. Other spreading-activation models have usually addressed this problem by using direct inhibitory connections between competing concepts (e.g. between meanings of *star* in Waltz & Pollack's network in Figure 2). For inferencing tasks, however, the inhibitory connections that these networks use are usually semantically unjustifiable and combinatorially explosive. The biggest problem, however, is that they are *winner-take-all networks*, acting to kill the activations of input interpretations that do not win the competition. This becomes a problem when a new context arises that makes an alternative interpretation more plausible. With the activations of the alternative interpretations killed by the inhibition from the false winner, it is exceedingly difficult for the activation from the new context to revive the correct one. The automatic reinterpretation capabilities of the networks are thus sabotaged.

In contrast, REMIND, like ROBIN, has no inhibitory links between competing concepts. It instead uses a group of units which act as a global inhibition mechanism. These *global inhibition* units serve to inhibit by equal proportions (normalize) all concepts in the network when their average activation becomes too high. The concepts in the network are thus free to keep an activation level relative to the amount of evidence in their favor. Global inhibition nodes are similar to the regulator units used by Touretzky & Hinton (1988), except that their regulator units are *subtractive inhibitory*, subtracting a constant amount of activation from all nodes and implementing a winner-take-all network, while REMIND's global inhibition nodes are *short-circuiting inhibitory*, controlling the spread of activation, but leaving *relative* values of evidential activation unchanged.

As opposed to driving the losers activations down to 0 using winner-take-all networks, REMIND's short-circuiting global inhibition mechanism allows all concepts in the network to hold a level of evidential activation relative to the amount of evidence in their favor. Letting losing interpretations keep activation proportional to their evidence enables REMIND to easily perform reinterpretation. When new context enters the network that favors an alternative interpretation over a previous one, it boosts the new interpretations relative levels of evidential activation — often being enough to cause the new interpretation to become most highly-activated. This occurs in **Hiding Pot**, in which the evidence from **P1** (*John put the pot inside the dishwasher)* initially favors Cooking-Pot. However, after evidence from the inferences of **P2** (*the police were coming)* is introduced, Marijuana's activation increases enough so that the network reinterprets *pot* as Marijuana.

3.7 Elimination of Crosstalk: Interaction of Signature and Evidential Activation

The use of signatures and evidential activation is a partial solution to the variable binding and inferencing problems of structured connectionist networks. However, the highly complex, overlapping, and ambiguous knowledge needed for language understanding (and eventual memory retrieval) requires more than simple integration of a variable binding mechanism and a standard evidential spreading-activation network. In particular, the problem of *crosstalk* inherent to all spreading-activation networks makes it crucial that the two paths of activation interact so that the dynamic variable bindings in the network control and channel the spread of activation (Lange, 1992). This section gives a brief overview of some of the problems of crosstalk for language understanding and how REMIND solves them. The reader interested mainly in memory retrieval may skip to the next section.

Crosstalk occurs when activation spreads from one area of a network into another area when it should not. This often provides unsupported activation evidence to a subset of network nodes and therefore disrupts the contextual disambiguation of the network. This can especially become a problem as networks get bigger and begin to have larger areas activated from inferences, as in REMIND. As an example, consider the following sentence:

John ate some rice before he went to church on Sunday morning. **(Church Service)**

The most probable interpretation of **Church Service** is that John had rice for breakfast before he went to attend services at church ($Church-Service). However, in a normal spreading-activation network, crosstalk from the combined activity of Rice and Church can cause $Wedding to become more highly-activated than $Church-Service, since $Church-Service would only receive evidence from Church. This is an example of crosstalk from *logically-unrelated inferences*. In general, Rice in the context of a Church should provide evidence for $Wedding. However, in the case of **Church Service**, Rice should not lead to the inference that a $Wedding is occurring, because the Rice is being eaten and not thrown.

One of the main potential sources of crosstalk in spreading-activation networks is that of *spurious*, or *logically-impossible* inferences, which is also a problem in many marker-passing systems. In **Church Service**, an inference path connecting the Ingesting of the Rice to $Wedding would be spurious, because the actions of eating and going to church are not causally related. As another example, consider the sentence:

After Bill put the omelet on the stove, he put the bowl inside the dishwasher. **(Cook-and-Clean)**

The most likely interpretation of **Cook-and-Clean** is that after Bill cooked his omelet, he put the bowl in the dishwasher so that he could clean it. Inside-Of-Dishwasher with Bill as the Actor and Bowl as the Object should be the winning refinement of Inside-Of. However, if Inside-Of-Stove (a refinement of Inside-Of leading to cooking inferences) is allowed to combine activation from Inside-Of with that from Stove, then Inside-Of-Stove could become more activated than Inside-Of-Dishwasher. The network might then might make the impossible inference that Bill was trying to cook something in the bowl even though he put it in the dishwasher.

REMIND's network structure controls such spurious, logically-impossible inferences by enforcing *selectional restrictions*, or binding constraints, on role-fillers. Selectional restrictions on role-fillers are defined in the knowledge base (e.g. Table 2), and encode knowledge such as that only Cooking-Utensils and Eating-Utensils are typically cleaned in dishwashers and that the Location of a dishwasher cleaning is a Dishwasher. The network structure enforces selectional restrictions by stopping activation from spreading to frames and roles whose selectional restrictions are violated. To perform this, each connection between binding units is actually a multiplicative connection that is gated by another unit calculating whether the signature is a legal one (cf. sigma pi units in Rumelhart, Hinton, & McClelland, 1986). The effect of this selectional restriction structure can be seen in the lower part of Figure 5. Here the network has recognized that Marijuana (6.8), the signature on the left binding node of Inside-Of's Object, violates Inside-Of-Dishwasher's Object's selectional restrictions. The gated connection (not shown) therefore does not allow Marijuana's signature to propagate to Inside-Of-Dishwasher's Object binding node, since only Cooking-Utensils and Eating-Utensils are cleaned in dishwashers. However, the network calculates that Cooking-Pot (9.2) does match the selectional restrictions. Cooking-Pot is allowed to propagate as a possible Object of Inside-Of-Dishwasher (Figure 5). Though not shown, the network recognizes selectional restriction violations basically by having units that compare the bindings signatures to those of the expected signatures.

In other cases, the role-fillers selectional restrictions on a frame are completely violated. For instance, Inside-Of-Stove and Inside-Of-Carwash are impossible interpretations for **Hiding Pot** and **Cook-And-Clean**, because the pot was put inside of a dishwasher and not a stove or car wash. In these cases, signature bindings interact with evidential activation so that the violated frames are completely stopped from receiving activation. Thus, selection restrictions dramatically reduce the number of spurious, logically-impossible inference paths generated by the propagation of signatures — and thereby allows REMIND to avoid one of the major pitfalls of simple marker-passing systems.

Another basic problem of structured connectionist networks is using the dynamic bindings of case role information to perform lexical disambiguation. For example, normal structured networks cannot distinguish between the sentences *The astronomer saw the star* and *The star saw the astronomer*. Signatures partially solve this problem by allowing the network to represent the bindings of the two sentences differently and keep track of who is seeing whom. However, if these bindings do not have an effect on the spread of evidential activation, then they do not help disambiguate between the meanings of *star*. For example, the extended Waltz and Pollack (1985) network of Figure 1 disambiguates the word *star* in *The astronomer saw the star* to a Celestial-Body, because Celestial-Body receives activation through hard-coded connections from both Astronomy and Sees Object. Unfortunately, when presented with *The star saw the astronomer*, the network does almost exactly the same thing. Celestial-Body again ends up with more activation than Movie-Star, because it is still receiving activation from the hard-coded connection from See's Object. This occurs even though Celestial-Body is not bound to See's Object in this story (see analysis in Lange, 1992). This is an example where the default, hard-coded case role connections that work in the general case can fail catastrophically in specific instances where the actual variable bindings are known.

ROBIN and REMIND solve this problem and use dynamic role-binding information to perform lexical disambiguation by using gated connections that give them a temporary *virtual structure* specific to the network's current dynamic bindings. These connections feed evidential activation back from frames *only* to concepts that are actually bound to their roles with signature activation. For example, in the case of *The star saw the astronomer*, only the signature of Movie-Star reaches the Actor role of See. Celestial-Body does not, because it violates See's selectional restrictions (celestial bodies have no eyes, and cannot see). Because frames only feed evidential activation back to the objects that are bound to their roles, Movie-Star therefore receives evidential activation from See's Object. Celestial-Body does not, and so Movie-Star wins as the interpretation of *star*.

The combination of selection restrictions and the channeling of evidential activation through the virtual role-binding structure solve the problems of crosstalk exemplified by **Cook-And-Clean** and **Church Service**. In **Cook-And-Clean**, the Inside-Of-Stove and stove cooking frames that might otherwise have won do not receive signature or evidential activation because their selectional restrictions are violated. In **Church Service**, the evidential activation from Rice goes to the eating frame that it is bound to and to the breakfast frames those bindings reach, and not (aside from a small amount of biasing activation) to the $Wedding frames. In both cases REMIND's structure avoids normal spreading-activation networks' crosstalk problems and allows it to arrive at the most plausible interpretation. REMIND's virtual structure is also often key to disambiguating more complex stories such as **Hiding Pot**. As shown in Figure 6, only Cooking-Pot reaches the $Dishwasher-Cleaning frames and receives evidential activation from them. Similarly, only Marijuana reaches and receives evidential activation from the Police-Capture frames. Thus, frames supported by contextual evidence of the network's inferences and bindings become more activated than their competitors, and so are more likely to be chosen as part of REMIND's interpretation of its input.

Finally, it is important to note that the evidential activation of the network also affects the spread of signatures. Signatures stop spreading to frames when the frames' evidential activation drops below threshold. This stops the network from making an infinite number of inferences — forward inferences are made, but only as far as there is support from context. For example, when input for *John put the pot inside the dishwasher* (**P1**) is presented to the network, evidential and signature activation reaches the Clean frame, but then drops below threshold and stops spreading. While **P1** provides enough evidence to infer that John might have put the pot inside the dishwasher to clean it, there is not enough evidence to make any further inferences as to why he would want to clean it. Further inferences, such as that John might have wanted to clean the pot to get ready for a Dinner-Party (as in **Dinner Party**), require additional evidential activation from the input, such as the convergent evidence from inferences for *company was coming*.

In summary, REMIND is able to avoid most crosstalk problems because of its structure of units and gated connections that allow signature bindings to control the spread of activation. Spurious inferences are avoided because activation only spreads to inference paths that are logically-possible interpretations of the input. Highly unlikely inferences are not made because signatures spread only to concepts that have evidential activation. And most importantly, the virtual structure created by signatures allows it to combine evidence as if the network was hand-built for the current bindings. Because of these interactions between evidential and signature activation, crosstalk is avoided, and REMIND is influenced only by contextually-appropriate evidence. Thus, the interpretation REMIND constructs of its cues is influenced only by its input activation, the biases of its connection weights, and its inferences.

4. Memory Retrieval

In REMIND, memory retrieval occurs automatically as a side-effect of the spreading-activation understanding process. Representations of previously-understood episodes are connected directly to the same semantic network that understood them in the first place. This direct form of indexing causes episodes that share conceptual similarities with the cue to become active as REMIND interprets the cue.

4.1 Representation of Long-Term Episodes

Whereas the general world knowledge and inference rules used to initially build REMIND's networks are hand-coded, REMIND is not given any information about the particular episodes it is going to understand and store in long-term memory. The representations used for these target episodes are created entirely by REMIND's spreading-activation understanding process. Input for each episode's text is presented to the network, which then infers an interpretation of it by the spread of signature and evidential activation. Next, units and connections are added (by hand) to store the episode's entire resulting interpretation in REMIND's long-term memory. Accordingly, each episode's representation includes all aspects of its interpretation, from its disambiguated surface features (such as the actors and objects in the story) to the plans and goals that REMIND inferred that the actors were using.

To determine the symbolic representation used to store an episode in memory, the state of the network is examined by hand to determine the interpretation it has settled on. As described previously, the network's interpretation is the most highly-activated path of frames and their role-bindings. For example, the representation of **Hiding Pot** that the network would store in long-term memory would include all of the instantiated frames and their disambiguated role-bindings in the dark gray area of Figure 6, representing the inferred interpretation of John hiding his Marijuana from the police to avoid being arrested.

As a complete example, consider how **Dirty Magazine** (*Billy put the Playboy under his bed so his mother wouldn't see it and spank him)* is processed and stored in the network as a memory episode. First, signature and evidential activations representing its phrasally-analyzed input are clamped to the network to start the understanding process. The actual phrasally-analyzed input given to the network for **Dirty Magazine** is:

```
(Phrase <Subject put Direct-Object under Indirect-Object>
   (Subject Billy)  (Direct-Object Playboy) (Indirect-Object bed))
(Phrase <Subject see Direct-Object>
   (Subject Mother) (Direct-Object Playboy))
(Phrase <Subject spank Direct-Object>
   (Subject ?) (Direct-Object Billy))
```

Possessives *(his)*, connectives *(so, would)*, and negations *(not)* are not included in the phrasally-analyzed input given to REMIND. The above input for **Dirty Magazine** could therefore be more accurately be described as: *Billy put Playboy under bed. Mother see Playboy. <Somebody unmentioned> spank Billy.* REMIND is left to infer the relations between the actions described by the individual phrases itself.

As described earlier for **Hiding Pot**, the input is presented to the network by clamping the evidential activations of the input's phrase and word nodes to 1 and clamping the binding units of the phrases' roles to the signatures of their bindings' word meanings. Activation then spreads through the network to infer and disambiguate an interpretation of the input. As in **Hiding Pot**, the network infers that somebody is hiding something (Avoid-Detection) and that it is blocked from sight (Block-See). Here, however, the inferred signatures show that it is Billy hiding a Playboy-Magazine rather than John hiding Marijuana.

The entire representation inferred for **Dirty Magazine** is shown in Figure 8. The ".1" after each frame name (e.g., Transfer-Under.1 and Bed.1) indicate that they are specific instantiations of concepts in the first episode processed by the network. For example, **Dirty Magazine**'s interpretation includes a surface action inferred directly from the first phrase *(Billy put the Playboy under the bed)*, i.e. that an instance of Transfer-Under, Transfer-Under.1, had occurred. The Actor of this Transfer-Under.1 is Billy.1, the Object is Playboy-Magazine.1, and the Location is Bed.1. The inferred representation also includes instantiations of more distant frames used by the network to understand **Dirty Magazine**, such as Avoid-Detection.1 and Block-See.1. As in **Hiding Pot**, the network's representation of **Dirty Magazine** also includes the possibility of a Punishment taking place, as it inferred for **Hiding Pot** (not shown in Figure 6) — though in this case the refinement of Punishment.1 is a Spank rather than an In-Jail. These similarities make **Dirty Magazine** a likely candidate for reminding when the network is presented with **Hiding Pot** as a cue.

It is important to note that each episode's representation also includes all of the simple bridging inferences that were necessary to make the plan/goal analysis. Here the bridging inferences for **Dirty Magazine** include that the Playboy was Under.1 the bed, that Billy possessed the Playboy (Possess-Obj.1), that the salient refinement of this possession was that it was possession of a naughty object (Possess-Naughty-Obj.1), and so on.

Once the full interpretation for an episode has been determined, units and connections representing this interpretation are hand-coded into the network's long-term memory. For **Dirty Magazine**, the units added include (a) nodes representing each instantiated frame of its interpretation in Figure 8 (e.g., Billy.1, Playboy-Magazine.1, Avoid-Detection.1, and Possess-Obj.1), (b) units to represent their roles, and (c) a unit to stand as a place holder for the entire episode (e.g., Episode.1). These units are then connected to their corresponding local elements in the normal evidential semantic network. They are also interconnected to encode their role-bindings and which episode they are part of.

Figure 9 shows an example of the units and connections that are added to the network to represent episodes. The figure shows a simplified part of the network's evidential layer after several episodes have been understood and added to long-term memory. The gray units in the figure are the normal semantic conceptual units originally in the network, including the conceptual units for frames Possess-Obj and Possess-Naughty-Obj and units in part of the physical object refinement (is-a) hierarchy. At this stage, two episodes have been processed that include Possess-Obj or Possess-Naughty-Obj as part of their interpretation: **Dirty Magazine** (Episode.1), and *Betty wanted to smoke a cigarette, so she put it on top of the stove and lit it* (**Cigarette Lighting**; Episode.4). **Cigarette Lighting**'s interpretation includes an instance of Possess-Obj because the network inferred that Betty must have possessed the cigarette to light it.

The white units in Figure 9 show some of the units added to the network to encode **Dirty Magazine** and **Cigarette Lighting**. For each episode, there is a single *episode unit* serving to represent and group all of its elements together, such as Episode.1 and Episode.4 in Figure 9. In addition, there is an *episode instance unit* representing each el-

```
(Instance Transfer-Under.1       0.61
        :Roles (Actor Billy.1)
              (Object Playboy.1)
              (Location Bed.1)
        :Phrase <S put DO under IO>.1
        :Refinement-Of       Do-Action-To-Obj.1
        :Results-In          Under.1)
(Instance Under.1                0.46
        :Roles (Planner Billy.1)
              (Object Playboy.1)
              (Location Bed.1)
        :Refinement          Under-Opaque.1
        :Result-of           Transfer-Under.1)
(Instance Under-Opaque.1         0.32
        :Roles (Planner Billy.1)
              (Object Playboy.1)
              (Location Bed.1)
        :Refinement-Of       Under.1
        :Results-In          Block-See.1)
(Instance Block-See.1            0.28
        :Roles (Planner Billy.1) (Object Playboy.1)
        :Result-Of           Under-Opaque.1
        :Blocks              See-Object.1
        :Plan-For            Avoid-Detection.1)
(Instance Avoid-Detection.1      0.10
        :Roles (Planner Billy.1)
              (Object Playboy.1)
        :Plan                Block-See.1)
(Instance See-Object.1           0.98
        :Roles (Actor Mother.1) (Object Playboy.1)
        :Phrase              <S see DO>.1
        :Precondition        Proximity-Of.1
        :Blocked-By          Block-See.1
        :Refinement          Guardian-See-Naughty)
(Instance Proximity-Of.1         0.48
        :Roles (Actor Mother.1) (Object Playboy.1)
        :Precondition-For    See-Object.1
        :Result-Of           Transfer-Self.1)
(Instance Transfer-Self.1        0.15
        :Roles (Actor Mother.1) (Object Playboy.1)
        :Results-In          Proximity-Of.1)
(Instance Do-Action-To-Obj.1     0.31
        :Roles (Actor Billy.1) (Object Playboy.1)
        :Refinement          Transfer-Under.1
        :Implies             Possess-Obj.1)
(Instance Possess-Obj.1          0.10
        :Roles (Actor Billy.1) (Object Playboy.1)
        :Refinement          Possess-Naughty-Obj.1
        :Implied-By          Do-Action-To-Obj.1)
(Instance Possess-Naughty-Obj.1 0.32
        :Roles (Actor Billy.1) (Object Playboy.1)
        :Refinement          Naughty-Committed.1
        :Refinement-Of       Possess-Obj.1)
(Instance Naughty-Committed.1 0.72
        :Roles (Actor Billy.1)
              (Evidence Playboy.1)
        :Precondition-For    Guardian-Know-Naughty.1
        :Refinement-Of       Possess-Naughty-Obj.1)
(Instance Guardian-See-Naughty.1 0.79
        :Roles (Actor Mother.1) (Evidence Playboy.1)
        :Refinement-Of       See-Object.1
        :Plan-For            Guardian-Know-Naughty.1)
(Instance Guardian-Know-Naughty.1 1.00
        :Roles (Actor Mother.1)
              (Object Billy.1)
              (Evidence Playboy.1)
        :Plan                Guardian-See-Naughty.1
        :Precondition        Naughty-Committed.1
        :Precondition-For    Guardian-Discipline.1)
(Instance Guardian-Discipline.1 0.89
        :Roles (Actor Mother.1)
              (Object Billy.1)
              (Evidence Playboy.1)
        :Precondition        Guardian-Know-Naughty.1
        :Results-In          Spank.1)
(Instance Spank.1                0.71
        :Roles (Actor Mother.1) (Object Billy.1)
        :Phrase              <S spank DO>.1
        :Result-Of           Guardian-Discipline.1
        :Refinement-Of       Punishment.1)
(Instance Punishment.1           0.38
        :Roles (Actor Mother.1) (Object Billy.1)
        :Refinement          Spank.1)
(Instance Unhappy-Punished.1  0.19
        :Roles (Planner Mother.1) (Object Billy.1)
        :Refinement-Of       Unhappy.1
        :Result-Of           Punishment.1)
(Instance Unhappy.1              0.14
        :Roles (Planner Mother.1) (Object Billy.1)
        :Refinement          Unhappy-Punished.1)
```

Figure 8. Interpretation inferred by spread of signature and evidential activation through the network for *Billy put the Playboy under his bed so his mother wouldn't see it and spank him* (**Dirty Magazine**). Frame names end in ".1" (e.g. Transfer-Under.1) to indicate that they are instantiations of concepts inferred in the first episode stored in the network. Numbers following instance names are final evidential activations of the frame.

ement of the episode's interpretation. For **Dirty Magazine**, there is an episode instance unit for Billy.1, Playboy-Magazine.1, Possess-Obj.1 and Possess-Naughty-Obj.1, along with units (not shown) representing all of the other elements of its representation. These episode instance units are connected both to the general semantic concept of which they are an instantiation (e.g., Billy.1 is connected to Billy) and to the episode unit of which they are part (e.g., Episode.1 for **Dirty Magazine**'s elements). Furthermore, each episode instance is connected to units representing its roles (e.g., the Actor and Object unit for Possess-Obj.1), which are in turn connected to the concepts that were bound to them (e.g., Possess-Obj.1's Actor is connected to Billy.1, and its Object is connected to Playboy-Magazine.1). The rest of the interpretation of each episode (e.g., in Figure 8) is encoded similarly with units and connections that represent all of its other instantiated frames and elements.

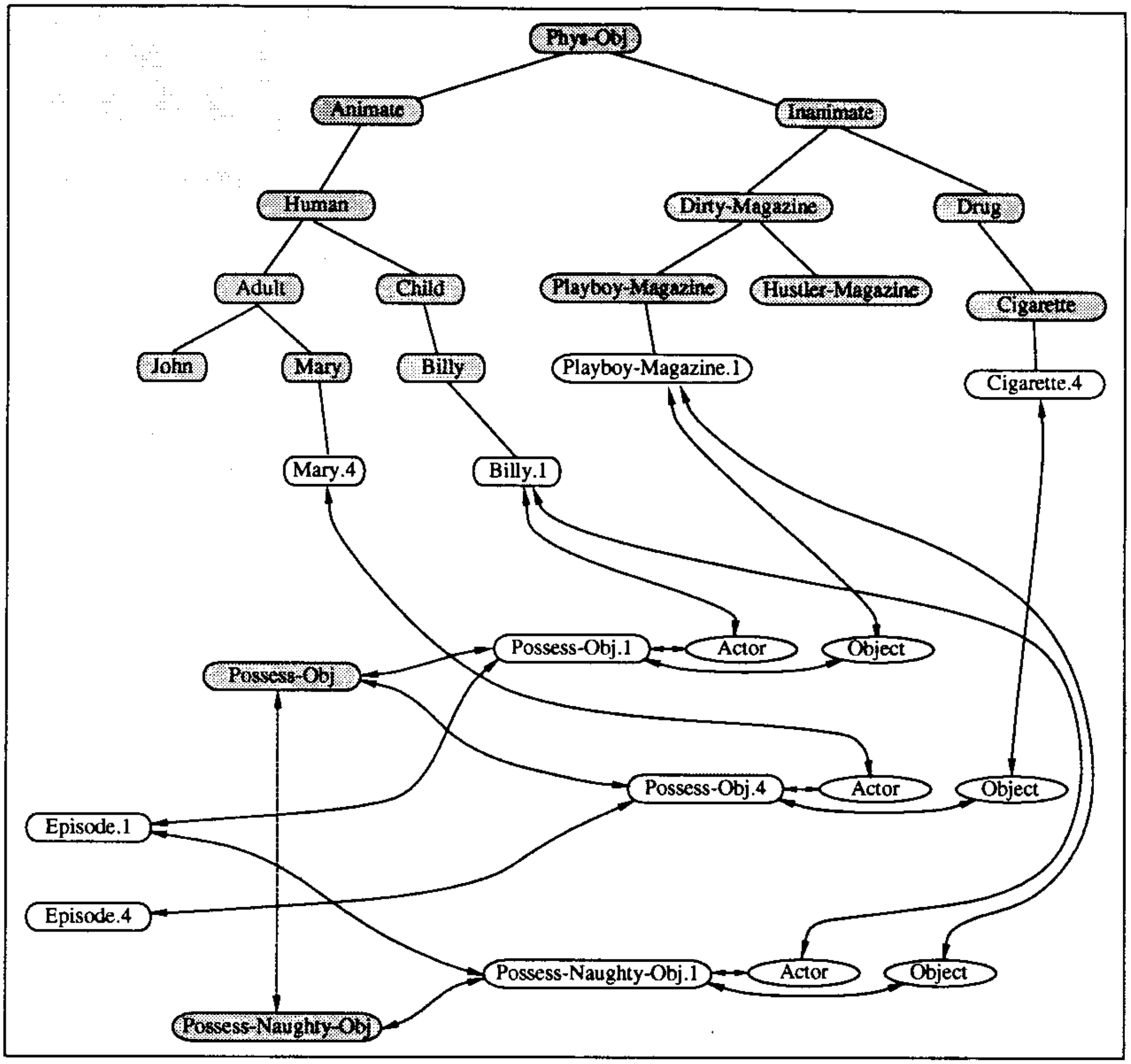

Figure 9. Encoding of Possess-Obj and Possess-Naughty-Obj instances for Episode.1 (**Dirty Magazine**) and Episode.4 (**Cigarette Lighting**). Gray units are pre-existing conceptual nodes. White units are nodes added to represent the episodes.

As can be seen, REMIND's method of encoding its episodes is different from that of many memory retrieval and case-based reasoning models. Episodes in REMIND are not indexed under any one knowledge structure or important groups of knowledge structures. They are instead indexed under every concept that was an aspect in understanding them in the first place. These concepts include both the surface features of the text (such as its direct disambiguated word and phrase meanings) and the abstract inferences that make up the plan/goal analysis of the episode. As will be discussed later, this fully dispersed form of indexing has important implications for the kinds of remindings that the model produces.

4.2 Detailed Connectivity of Episodic Units

Unlike the inferencing and understanding portion of the network, the units and connections representing long-term memory episodes reside entirely on the evidential layer of the network. Because the bindings of each individual long-term memory episode are fixed once an episode has been understood and remembered, the bindings can be encoded by direct connections between role units and the elements that are bound to them. Thus, long-term memory episodes do not need the more complex inferencing structure that holds and propagates dynamic signature bindings.

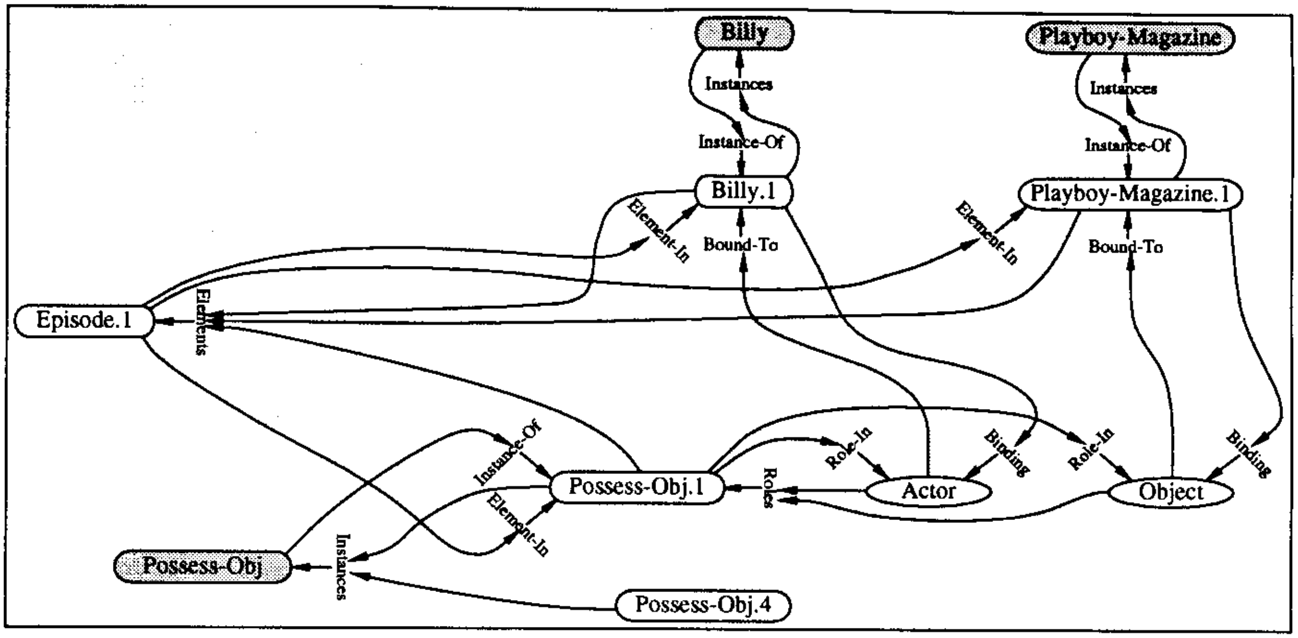

Figure 10. Detailed view of units added to represent Possess-Obj.1 of Episode.1. Labels next to nodes (e.g. Instances, Instance-Of) represent their input branch units.

An example of the full set of units and connections used to encode each frame instance in an episode is shown in Figure 10. As in the normal conceptual units of REMIND, each incoming connection to a concept unit from other concepts goes through an *input branch unit.* Input branch units are analogous to the *input sites* described by Cottrell and Small (1982). They serve both to specify the relationships concepts hold to each other and to control the spread of evidential activation.

Episode instance units have four input branches: an Instance-Of branch, an Element-In branch, a Roles branch, and a Bound-to branch. For example, Possess-Obj.1 has three input branch units in Figure 10, (a) an Instance-Of input branch that receives activation from Possess-Obj, the concept it is an long-term instance of, (b) an Element-In input branch that receives activation from the episode it is a part of (i.e., Episode.1), and (c) a Roles input branch that receives activation from each of its role units (i.e., its Actor and Object units). The Instance-Of and Element-In branch units calculate their activation as the sum of their single input, since any one instance can only be an instantiation of one general semantic frame. Likewise, any one instance can only be an element in one episode. The Roles input branch, on the other hand, calculates its activation as the *average* of its role unit inputs, so that frames with multiple roles do not become the most active simply because they have more role-bindings than simpler frames. Episode instance units also have a Bound-To input branch that has connections from any of the episode roles they are bound to. For example, Billy.1 is bound to the Actor roles of Possess-Obj.1 and Possess-Naughty-Obj.1 (amongst others), and so receives activation from those roles through its Bound-To branch. Bound-To input branches calculate their activation as the *maximum* of their inputs, so that they receive activation from the most active role they are bound to.

Episode role units, such as Possess-Obj.1's Actor and Object roles, have two input branches: a Role-In branch and a Binding branch. Role-In branches receive activation from the episode instance unit of which the episodic role is part. In Figure 10, Possess-Obj.1's Actor and Object roles therefore both have Role-In branches that receive activation from Possess-Obj.1. In contrast, Binding branches have a connection from the element that is bound to their role to represent its long-term role-binding. Since Possess-Obj.1's Actor role is bound to Billy.1, it therefore receives activation from Billy.1 through its Binding branch. Similarly, the Binding branch of Possess-Obj.1's Object role receives activation from Playboy-Magazine.1.

The episode units themselves have a single Elements branch that receives and sums up the activation of all elements that make up the episode. For example, Episode.1's Elements branch receives activation from Billy.1, Playboy-Magazine.1, Possess-Obj.1, and all the other episode instance elements of its interpretation. This serves two functions: to keep track of all the elements in each episode, and to cause episode units to become active when their elements are active.

Finally, all of the concept units in the normal evidential semantic network have an Instances input branch that is used to activate them from each of their instantiations in long-term memory. For instance, Possess-Obj has an Instances branch that receives activation from Possess-Obj.1, Possess-Obj.4 and all of its other episode instances (not shown). This is in addition to the normal conceptual input branches of the network (see Lange & Dyer, 1989). An Instances branch unit calculates its activation as the *maximum* of its inputs, because, in general, REMIND can only be reminded of one instantiation of a given frame at a time. An important effect of this activation function is that it also stops concepts that have been seen in many stories before (and therefore have a lot of instances) from dominating concepts that may have more unique (and therefore fewer) instantiations in memory.

All connections in Figure 10 have unit weight, with the exception of the connections from the episode units (e.g., Episode.1) to the Element-In branches of their episode instances, which have a weight of 0.05. The unit weights from semantic concepts to each of their episode instances make episode instances likely to become active when related concepts are active. On the other hand, the small weights from episode units to the Element-In branches of episode instance units cause their elements to become moderately primed when their episode is active, without becoming too active unless they share other similarities with the reminding cue.

4.3 The Process of Episodic Reminding

Retrieval in REMIND begins with presentation of an input cue to the network to be understood. Because episode instance units are connected directly to their corresponding concept units, they become active when the concepts they are instantiations of become activated by the understanding process. The more similarities an episode shares with the inferred interpretation of a cue, the more of the episode's instance units will become active. Episodes having a number of elements in common with the cue's interpretation therefore tend to become highly active. After the network settles, the episode with most highly-activated episode unit is retrieved.

Figure 11 shows an overview of part of the network after it has understood and encoded the eight different episodes shown in Table 3. The circled numbers above frame nodes in the figure indicate instantiations in different episodes that are connected to the frames. For example, the episode instance units for Possess-Obj.1 and Possess-Naughty-Obj.1 of Episode.1 (**Dirty Magazine**) shown in Figure 9 are indicated by the circled 1's above Possess-Obj and Possess-Naughty-Obj in Figure 11.

Notice that more specific frames tend to have fewer episode instances than less specific frames. This is to be expected, since specific knowledge structures pertaining to certain situations (such as a police search or a parent disciplining a child) represent events that are less frequently encountered than general knowledge structures about simple actions and states (such as being inside of something, or possessing an object). As an example, five of the episodes in Table 3 and Figure 11 (1, 2, 4, 6, and 7) inferred a Possess-Obj as part of their interpretation, but only one episode (1) involved a Possess-Naughty-Obj or Avoid-Detection. An important consequence of specific frames providing activation evidence for a smaller number of instances is that specific, contentful knowledge structures tend to be stronger reminding indices than general ones.

Now consider what happens when input for **Hiding Pot** is presented as a cue to the network. Evidential and signature activation spread through the network, dynamically instantiating the competing inference paths as described earlier. Figure 12 shows the activation levels of the eight episodes as activation spreads through the network. As can be seen, Episode.6 (*Barney put the flower in the pot, and then watered it)* initially becomes highly active because it shares a number of surface features with **Hiding Pot** for example, both involve a Transfer-Inside, both have humans, and Planting-Pot receives activation from the word *pot*. Similarly, Episode.2 (**Car Wash**) initially becomes active because of shared surface features with **Hiding Pot**. Episode-2's activation continues to climb when the Clean frame is inferred, since a Clean is part of **Car Wash**'s interpretation. However, as REMIND continues to process **Hiding Pot**, the hiding and punishment frames are inferred and become active. Eventually, Episode.1's (**Dirty Magazine**) activation climbs and wins because it shares the most surface *and* abstract features of any episode with **Hiding Pot**'s interpretation (see Figure 11). **Dirty Magazine** is therefore retrieved as the episode most similar to **Hiding Pot**.

Table 3: Episodes for which input was understood and stored in the network of Figure 11.

#	Episode Text	Phrasally-Parsed Input Given the Network	
1	*Billy put the Playboy under his bed so his mother wouldn't see it and spank him.* **(Dirty Magazine)**	(<S put DO under IO> (<S see DO> (<S spank DO>	(S Billy) (DO Playboy) (IO bed)) (S Mother) (DO Playboy)) (DO Billy))
2	*Fred put his car inside the car wash before his date with Wilma.* **(Car Wash)**	(<S put DO inside IO> (<S date DO>	(S Fred) (DO car) (IO carwash)) (S Fred) (DO Wilma))
3	*Jane shot Mark with a Colt-45. He died.*	(<S shot DO with IO> (<S died>)	(S Jane) (DO Mark) (IO Colt-45)) (S Mark))
4	*Betty wanted to smoke a cigarette, so she put it on top of the stove and lit it.*	(<S smoke DO> (<S put DO on top of IO> (<S lit DO>	(S Betty) (DO cigarette)) (S Betty) (DO cigarette) (IO stove)) (DO cigarette))
5	*The pleasure boat followed the whales to watch them.*	(<S followed> DO> (<S watch DO>	(S pleasure-boat) (DO whales)) (DO whales))
6	*Barney put the flower in the pot, and then watered it.* **(Flower Planting)**	(<S put DO inside IO> (<S watered DO>	(S Barney) (DO flower) (IO pot)) (DO flower))
7	*Mike was hungry. He ate some fish.*	(<S was hungry> (<S ate DO>	(S Mike)) (S Mike) (DO fish))
8	*Suzie loved George, but he died. Then Bill proposed to her. She became sad.* **(Sad Proposal)**	(<S loved DO> (<S died> (<S proposed to DO> (<S became sad>	(S Suzie) (DO George)) (S George)) (S Bill) (DO Suzie)) (S Suzie))

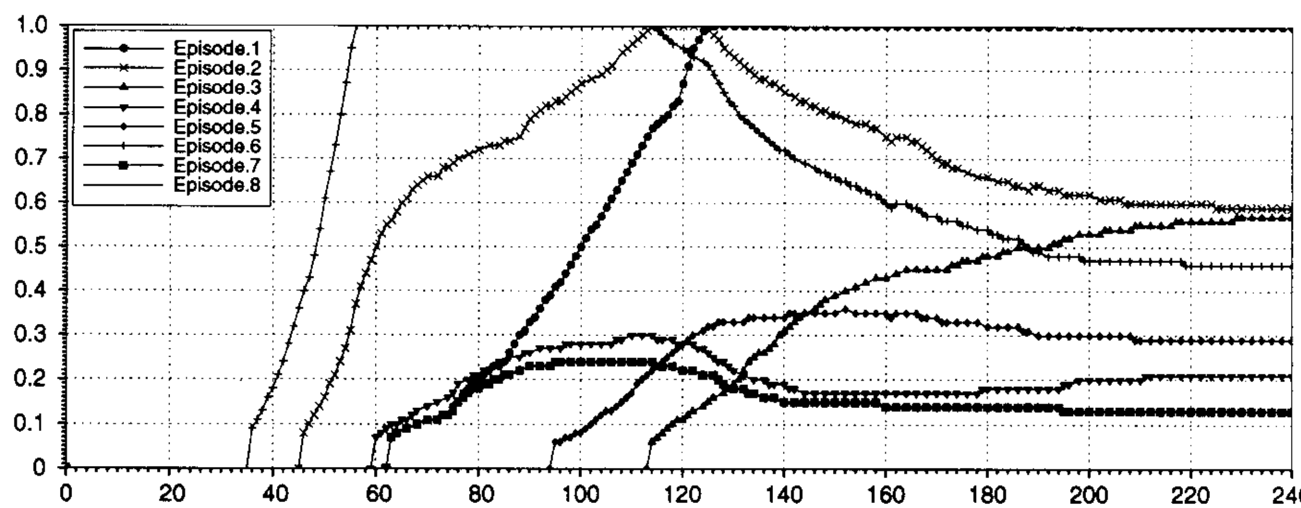

Figure 12. Evidential activations of episode units for eight episodes of Table 3 after presentation of **Hiding Pot**.

An explanation for why **Dirty Magazine** becomes the most-highly activated of the eight episodes can be seen in Figure 11. The gray boxes around nodes in Figure 11 indicate the final levels of evidential activation of the frames inferred for **Hiding Pot**. Of the eight episodes stored in the network, **Dirty Magazine** has the most instances of its interpretation shared with **Hiding Pot**'s final active interpretation (e.g., instantiations Avoid-Detection.1, Block-See.1, Punishment.1, and Possess-Obj.1). It therefore eventually becomes the most activated of the episodes.

Figures 13 and 14 show the activation levels of different elements in **Flower Planting** (Episode.6) and **Dirty Magazine** (Episode.1) as **Hiding Pot** is being understood. **Flower Planting** initially becomes the most active of the episodes because it shares a number of superficial similarities with the undisambiguated phrase *John put the pot inside the dishwasher*. As shown in Figure 13, it quickly receives activation when its element Planting-Pot.6 becomes activated from the Planting-Pot meaning of *pot* at about cycle 25. Additional activation is received when Transfer-Inside.6 becomes activated from Transfer-Inside at about cycle 40. The activation of Episode.6 con-

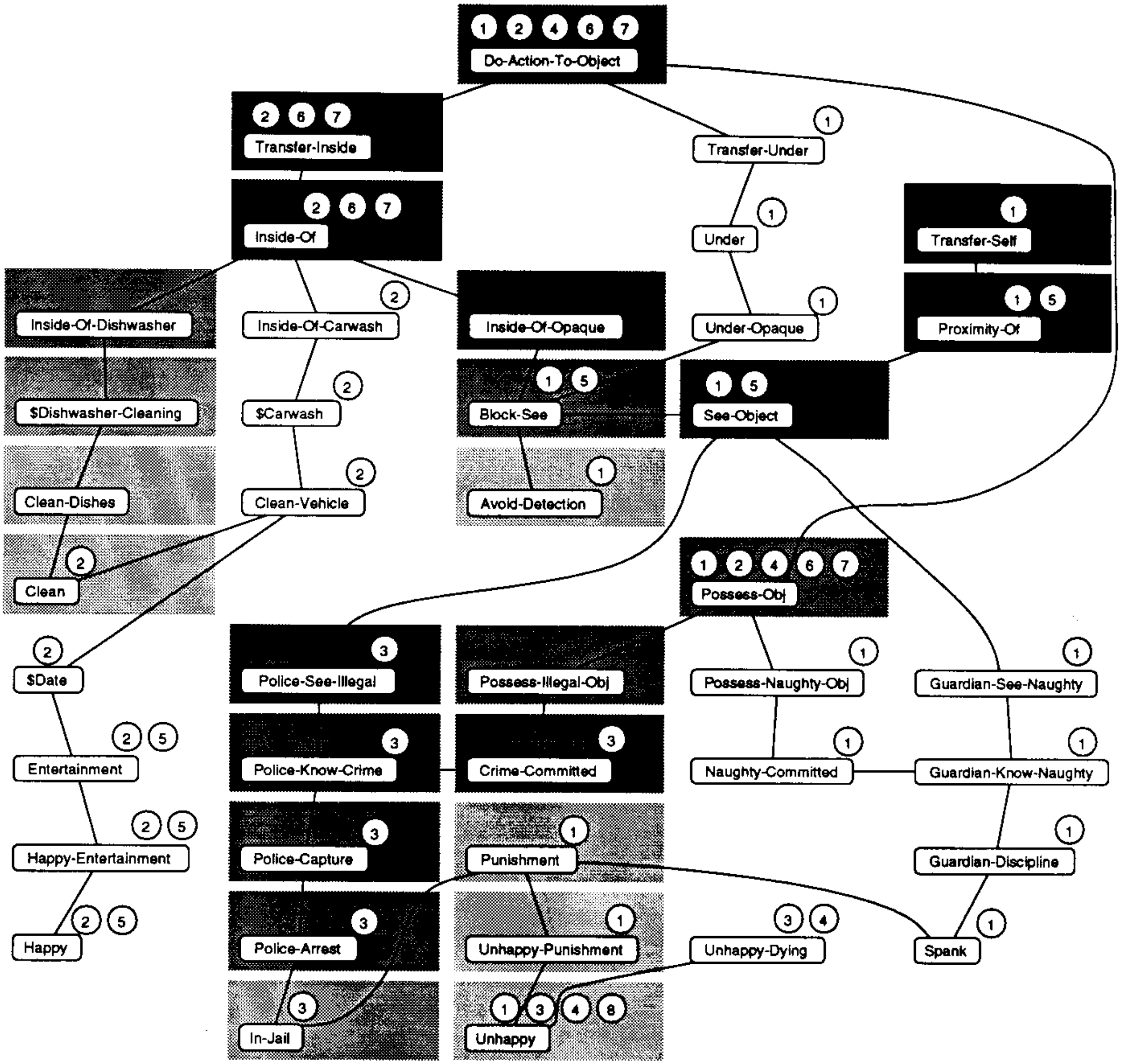

Figure 11. Overview of part of the network after activation has settled in processing of **Hiding Pot**. Gray boxes around nodes represent the level of evidential activation on the frame concept nodes (darker = higher activation, no box = no activation). Circles above frames indicate long-term instances connected to them. Numbers within circles indicate which episode the instance is part of in Table 3.

tinues to climb along with Planting-Pot.6 and Transfer-Inside.6, and gets even more activation when the network infers that the pot is Inside-Of the dishwasher, directly activating Inside-Of.6 at about cycle 50. That, however, is where **Flower Planting**'s similarities with **Hiding Pot**'s inferences end, except for a couple of other shared elements (Do-Action-To-Object.6 and Possess-Obj.6). For example, Inside-Of-Planting-Pot.6, part of **Flower Planting**'s interpretation, does not become significantly active because Inside-Of-Planting-Pot is never inferred by the network (because a pot inside a dishwasher violates its selectional restrictions).

Figure 14 shows the activation levels of **Dirty Magazine**'s elements as the network eventually infers the plans and goals of **Hiding Pot** that the two episodes share. One of the first similarities activated is the Possess-Obj.1 instantiation of Episode.1 (see Figure 9) after about cycle 60. This causes Episode.1's activation to climb above threshold, though its activation is still much lower than that of Episode.6 or Episode.2. However, as time goes on,

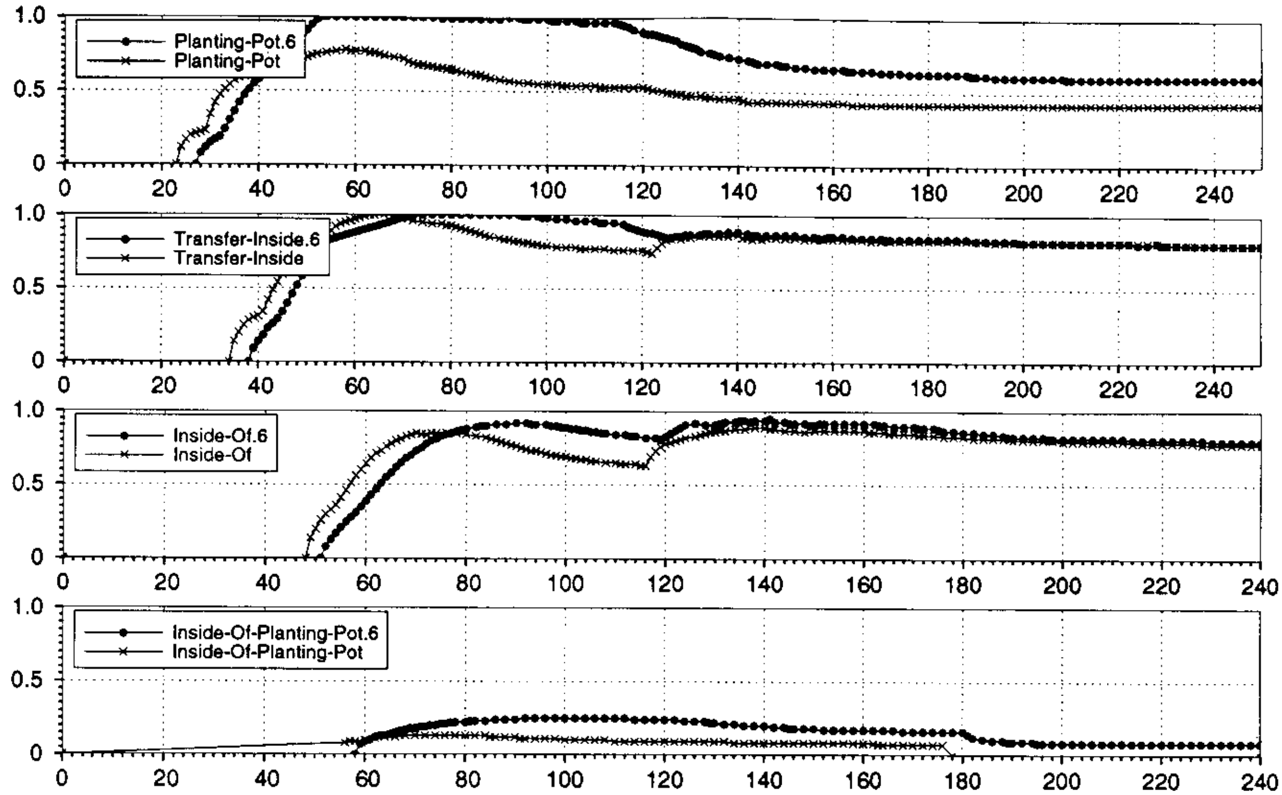

Figure 13. Activation of elements of Episode.6 (**Flower Planting**) after presentation of **Hiding Pot.**

Block-See, See-Object, Avoid-Detection, Punishment, and the other shared knowledge structures of **Dirty Magazine** are inferred by the spread of signatures and evidential activation for **Hiding Pot**, so that eventually the cumulative evidence from all the shared inferences causes Episode.1 to become the most highly activated episode at around cycle 170 (Figure 12).

Besides serving as an example of retrieval in REMIND, this example illustrates a number of important points about the model. The first point to notice is that even when the network settles, the losing episodes retain a level of evidential activation relative to the amount of evidence available for them, rather than being driven down to zero. As in the normal evidential semantic network, this is the result of controlling episodes activations through REMIND's global inhibition rather than normal mutual inhibition.

A second point of interest is that elements and episodes that are superficially similar to the cue tend to become activated *before* elements and episodes that are only abstractly related to the cue (through inferences). This is a direct result of the spreading-activation process, since activation and signature inferences reach closely-related concepts before they reach more distant concepts. An example of this was seen in Figure 12, where the superficially-related Episode.6 became activated before the more abstractly-related **Dirty Magazine**. As seen, however, the early activation of superficially-similar episodes does not stop thematically-similar episodes from winning if the thematically-similar episodes ultimately share more features and activation with the cue. Because all episodes retain their relative supported levels of activation, thematically-similar episodes such as **Dirty Magazine** can climb as inferences reach them and end up with the highest level of activation when the network settles.

Another important thing to note is that retrieval in REMIND is not all-or-nothing. As in human recall, REMIND often gets *partial* recall in which only subparts of the retrieved episode are activated. Parts of the retrieved episode distant from the current context of inferences may not become activated initially. Possess-Naughty-Obj.1, for example, only becomes partially primed (from its Element-In and Roles branches), because Possess-Naughty-Obj was never activated from the inferences made for **Hiding Pot**. The same is true for most of the other parts of **Dirty Magazine** that differ significantly from **Hiding Pot**'s interpretation (such as the Guardian-Discipline and Spank structures, which are relatively distant from anything in **Hiding Pot**). However, the primary actors and objects in

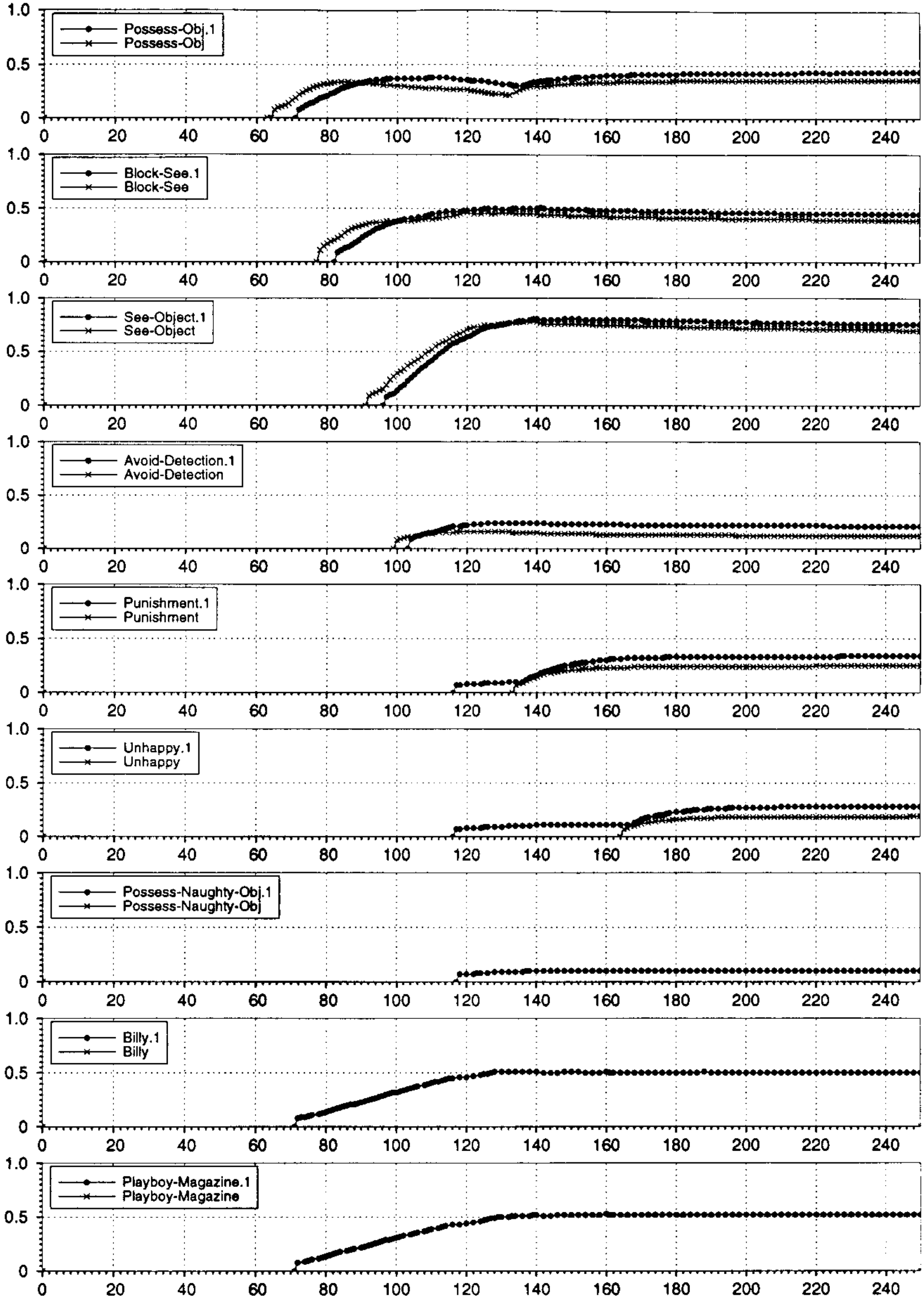

Figure 14. Evidential activation of elements of Episode.1 (**Dirty Magazine**) after presentation of **Hiding Pot**.

episodes, such as Billy.1 and Playboy-Magazine.1 in **Dirty Magazine**, do tend to become active because they play a part in so many of its roles.

5. Experiments and Discussion

REMIND has been implemented and tested in the DESCARTES connectionist simulator (Lange, Hodges, Fuenmayor, & Belyaev, 1989). The knowledge base used for the examples in this paper currently includes 206 distinct conceptual frames (e.g. Inside-Of, Avoid-Detection, Cooking-Pot) and 333 inference rules (e.g. R1 - R4). It has understood and retrieved the examples presented here (including **Hiding Pot**, **Dinner Party**, and the episodes of Table 3) and a number of other episodes of similar lengths and complexity.

In this section we discuss three simulations that illustrate (1) the importance of inferences and disambiguation on retrieval in REMIND, (2) the strong influence of superficial feature similarities on retrieval, and (3) the affect of episodic recall on the understanding process. We then compare REMIND to the ARCS and MAC/FAC models of general reminding. Finally, we discuss several directions we are exploring to extend the model.

5.1 Importance of Inferences

The retrieval of **Dirty Magazine** when **Hiding Pot** was presented as a cue to REMIND illustrates the importance of inferencing and disambiguation in the retrieval process. **Dirty Magazine** was retrieved over the more superficially similar **Flower Planting** and **Car Wash** episodes because **Dirty Magazine** and **Hiding Pot** were more thematically similar, since both involved somebody hiding something to avoid punishment. Without being able to infer these similarities, the model would not have been able to retrieve the most analogous episode.

Another example of how text comprehension affects REMIND's retrieval process is shown when two superficially-similar cues that have entirely different interpretations are presented to the network. On the surface, the cue *John put the pot inside the dishwasher because company was coming* (**Dinner Party**) is nearly the same as **Hiding Pot.** The only difference is that *company* is coming rather than *the police*. Analogical retrieval models such as ARCS (Thagard et al., 1990) and MAC/FAC (Gentner & Forbus, 1991) that do not have inferencing mechanisms would therefore have to predict that the same episode would be retrieved for both cues from amongst those shown in Table 3. This would occur for two reasons. First, the only surface feature difference between the two cues does not make a difference in overall surface similarity between the cues and any of the episodes from Table 3 — *company* and *police* are equally similar to all objects in these episodes. Second, the isomorphic structure (structural consistency) of the two cues is the same.

In contrast to ARCS and MAC/FAC, REMIND retrieves different episodes for **Dinner Party** and **Hiding Pot.** This is because REMIND interprets the two cues very differently. Whereas in **Hiding Pot** it appears John is trying to hide marijuana from the police, in **Dinner Party** it appears that he is trying to clean a cooking pot in preparation for company coming over. As such, **Dinner Party** is not likely to cause a reminding of **Dirty Magazine** and its hiding event. It seems more likely that **Dinner Party** would cause reminding of another episode involving somebody cleaning something in preparation for entertaining, such as the **Car Wash** episode (*Fred put his car in the car wash before his date with Wilma).*

This is exactly what happens in REMIND when episodic memory contains **Dirty Magazine, Car Wash**, and the other six episodes of Table 3. Figure 15 shows the evidential activations of the meanings of *pot* and some of the competing refinements of Inside-Of after REMIND is given input for **Dinner Party**. As when **Hiding Pot** is presented, Planting-Pot initially becomes highly activated because the input for *John put the pot inside the dishwasher* (**P1**) contains the word *pot*. After about cycle 60, REMIND infers Inside-Of-Dishwasher and Inside-Of-Opaque. Cooking-Pot's activation therefore starts to climb because of its unmatched activation from Inside-Of-Dishwasher and the other $Dishwasher-Cleaning frames. Input for *company was coming* is presented to REMIND at cycles 51-61. Spreading signature and evidential activation makes candidate inferences for *company was coming* starting from Transfer-Self, through Proximity-Of and then the Dinner-Party frames (representing somebody making dinner for somebody else). Activation eventually reaches the Clean-Dishes frame (a *precondition* for Dinner-Party). The cleaning frames were already active after the initial inferences from **P1**, so activation continues to spread and provide added support for the Inside-Of-Dishwasher refinement of Inside-Of at about cycle 130. Inside-Of-Opaque also gets added evidence at about cycle 140 from inferences through the Block-See frames from *company*

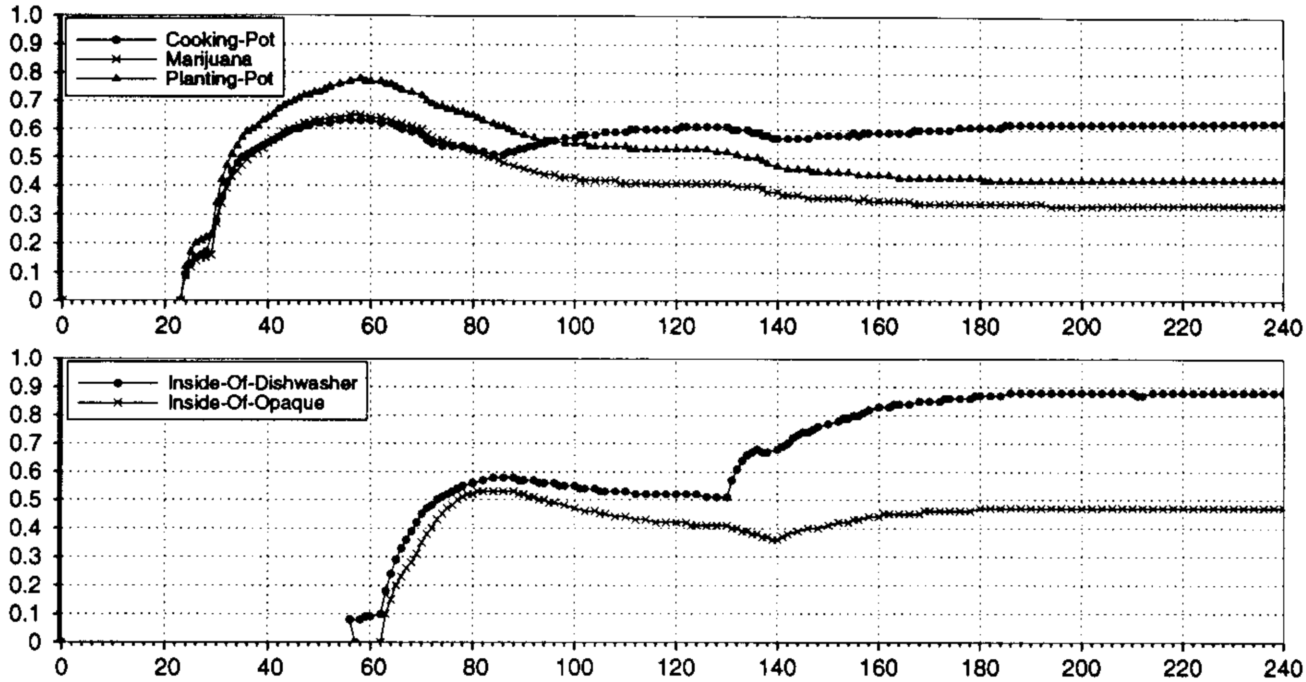

Figure 15. Evidential activations for meanings of *pot* and of competing refinements of Inside-Of after presentation of input for *John put the pot inside the dishwasher* (P1) at cycles 1 through 31 and *company was coming* (P2) at cycles 51 through 61 (**Dinner Party**)

was coming (since John could still have been hiding the pot). However, Inside-Of-Opaque does not receive enough activation to compete with Inside-Of-Dishwasher. Similarly, Marijuana never seriously competes with Cooking-Pot, because the Police-Capture and Police-See-Illegal frames that gave it unique activation in **Hiding Pot** are not inferred here (since Company does not match their selectional restrictions). The network's final interpretation of **Dinner Party** is therefore that John was trying to clean a Cooking-Pot to prepare for a Dinner-Party he was giving for Company.

Figure 16 shows the activation of the eight episodes and of two of **Car Wash**'s elements as **Dinner Party** is processed. As when **Hiding Pot** is presented, Episode.6 (Mike put the flower in the pot, and then watered it) is the first episode to become activated because of its surface similarities with *John put the pot inside the dishwasher*. Episode.2 (**Car Wash**) gets activated after about cycle 45 because of the shared inferences of Transfer-Inside and Inside-Of, and continues to gain more activation when Clean is inferred at around cycle 100. The activation of Episode.1 (**Dirty Magazine**) also starts climbing at around cycle 100 because the network is also activating the hiding and seeing structures as possible inferences for **Dinner Party**. At about cycle 120, the network infers that an Entertainment is being planned (the Dinner-Party). This activates Entertainment.2 and provides more evidence for **Car Wash**. At about the same time, the Cleaning frames (including Inside-Of-Dishwasher) get reinforcement from the *company was coming* inferences (cycles 100-140). This boosts the activations of many of the remaining elements of **Car Wash,** such as Clean.2. Accordingly, Episode.2's activation begins to dominate over Episode.1's (**Dirty Magazine**), which gradually loses support starting about cycle 150. Thus, REMIND retrieves the **Car Wash** episode when **Dinner Party** is presented as a cue. This demonstrates that even changing a single word in the cue (from *police* to *company)* can completely change the inferences and interpretations REMIND makes, and, consequently, the episode it retrieves.

5.2 Superficial Similarities

As mentioned earlier, human reminding seems to be affected strongly by superficial feature overlap between cues and memory episodes. Even when episodes exist in memory that are highly analogous to a cue (and hence useful for problem solving), people may instead get reminded of non-analogous episodes simply because they are more superficially similar. For example, stories about sharks eating people, such as **Killer Shark**, may remind people of other stories about sharks or other man-eaters devouring people, rather than thematically similar stories that do not

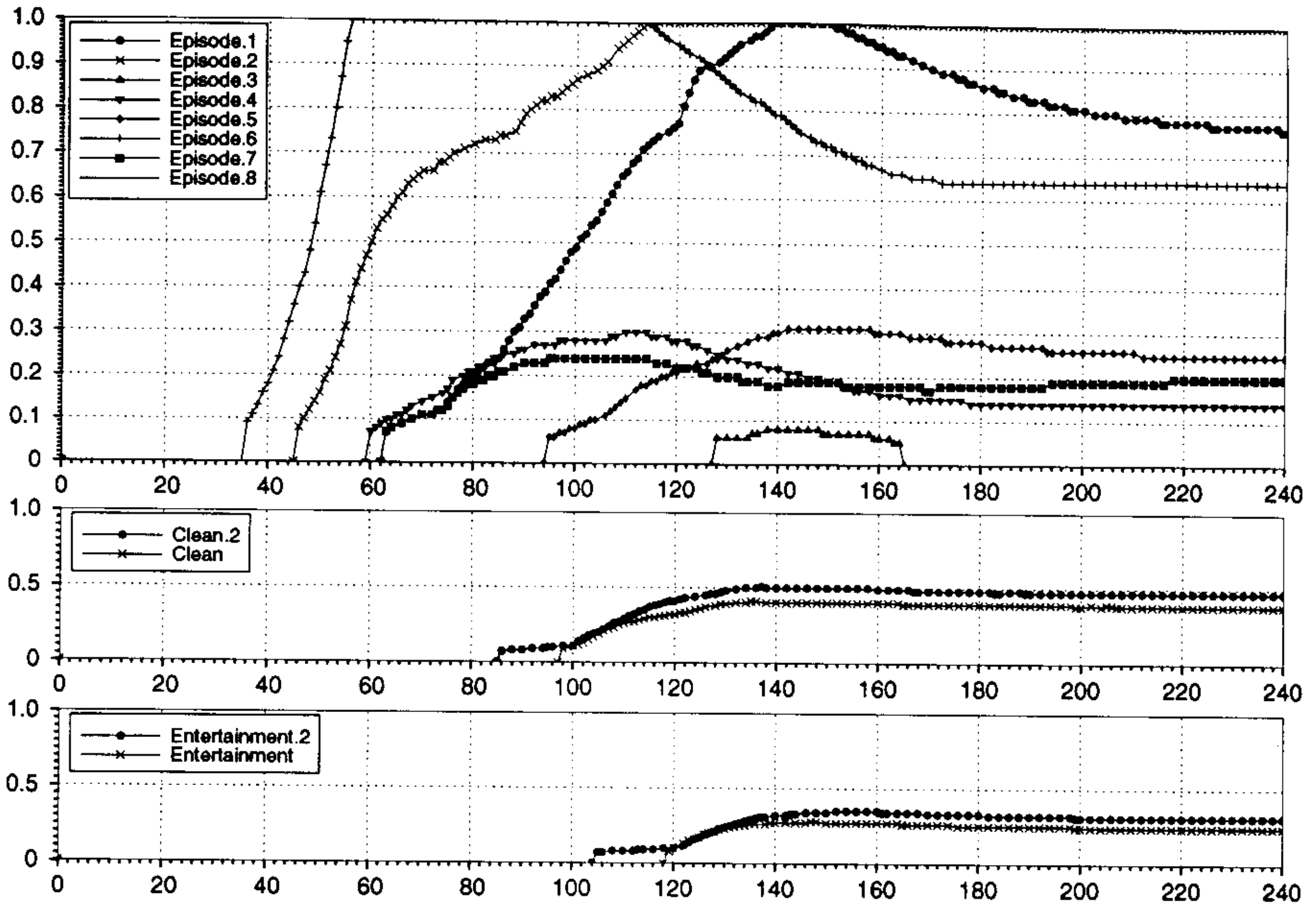

Figure 16. Activations of eight episodes of Table 3 and two of the elements of winning Episode.2 (**Car Wash**) after presentation of **Dinner Party**.

happen to involve sharks. Superficial remindings therefore often come at the expense of perhaps more valuable cross-contextual remindings.

This characteristic of reminding can be explained quite readily if episodes are remembered by storing all aspects of their interpretation, as in REMIND. Because REMIND stores all of the knowledge structures used in building an interpretation of the episode, from surface features to abstract inferences, it predicts that episodes that share many surface features with a cue are indeed likely candidates for reminding. For example, consider the sentence:

Cheech put the grass inside the bong because Chong was coming. (**Cheech and Chong**)

Cheech and Chong is an example of a superficially similar episode that can prevent retrieval of an analogous episode. Although **Cheech and Chong** is not analogous to **Hiding Pot**, the two episodes share a number of surface features. Both involve marijuana, marijuana being put inside of something, and somebody coming near. The plans and goals in the two episodes are completely different, however. In **Cheech and Chong**, the most probable interpretation is that Cheech was readying the marijuana to be smoked with his friend Chong. In **Hiding Pot**, of course, the network's interpretation was that John was hiding the marijuana so he would not be punished. So if **Cheech and Chong** is understood and stored in memory along with the eight episodes of Table 3, then **Dirty Magazine** is still the most *analogous* story in memory to **Hiding Pot**. **Cheech and Chong**, however, shares far more *total* features with **Hiding Pot**'s interpretation, and is therefore more likely to be retrieved by the model when **Hiding Pot** is presented as a cue.

To test this, input for **Cheech and Chong** was presented to the network to be understood and then remembered. The network disambiguated *grass* to Marijuana (instead of Lawn-Grass), and inferred an interpretation that Cheech put the marijuana inside a Marijuana-Bong to Light it for the Pot-Party that Chong was transferring himself to. **Cheech and Chong**'s interpretation was then added (as Episode.9) to the eight episodes already stored in the network.

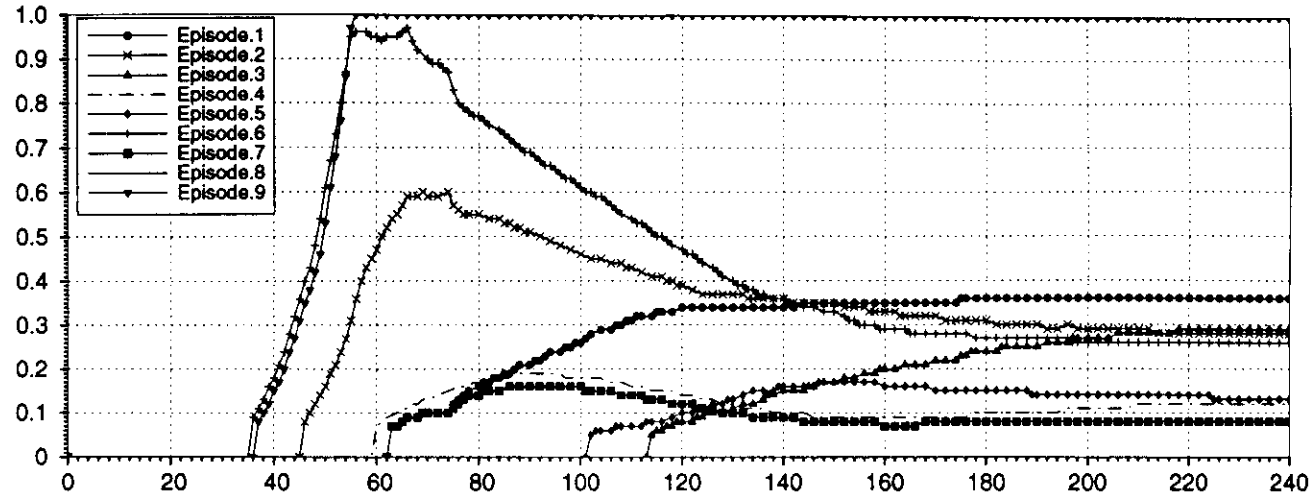

Figure 17. Activations of eight episodes of Table 3 and Episode.9 (**Cheech and Chong**) after presentation of **Hiding Pot**.

Figure 17 shows the evidential activations as **Hiding Pot** is being understood in the network containing **Cheech and Chong** and the eight other episodes of Table 3. Both **Cheech and Chong** (Episode.9) and **Flower Planting** (Episode.6) quickly become activated because of their surface similarities with the undisambiguated *John put the pot inside the dishwasher*. However, Episode.9's activation starts to dominate and Episode.6's starts to fall after *the police were coming* is presented to the network. This occurs because *the police were coming* adds more superficially similar activation to Episode.9's *Chong was coming*. Episode.6's activation drops further when Planting-Pot's activation falls as the network disambiguates *pot* to Cooking-Pot (the cleaning inferences) or Marijuana (the hiding and police capture inferences). Episode.9's activation, on the other hand, remains at its maximum (1.0) until the network settles. **Cheech and Chong** is therefore the episode retrieved for **Hiding Pot**. The other eight episodes end up with relatively little activation, since **Cheech and Chong** is so (superficially) similar to **Hiding Pot** relative to them. As might be expected, **Dirty Magazine** is the most highly activated of the remaining eight episodes, since it was the episode most similar to **Hiding Pot** before **Cheech and Chong** was remembered.

5.3 Effect of Reminding on Interpretation

REMIND only models how cues are understood and how episodes are consequently retrieved from memory. Unlike case-based reasoning models, it does not model how the information in those episodes can be used for analogical transfer or applied for problem-solving. One of the reasons this would be difficult for REMIND is that the *analogical inferences* that are made by case-based reasoning models are essentially equivalent to applying *novel* rules — e.g. applying a newly-mapped rule from a previous case to the bindings of the new case. Structured connectionist networks such as REMIND cannot currently represent such completely novel rules, because rules connecting concepts are hard-coded with units and links that cannot themselves be dynamically recruited. The possible extensions to signatures' representational and inferencing capabilities discussed in Lange (1992) might solve some of these problems. Eskridge (in press) and Barnden and Srinivas (in press) show that *hybrid* connectionist models that use complex symbolic abilities can perform analogical and case-based transfer from the cases they retrieve.

Though it does not reason from the episodes it recalls, REMIND's integration of the reminding and understanding processes shows that memory retrieval can have pragmatically interesting and useful effects on the understanding process. Episodes that become active during the understanding process feed activation back into the inferencing part of the network. This, in effect, can prime and bias the interpretation REMIND settles on for a given input. For example, consider the following example:

> *The star loved the plumber, but he was shot by a thief. Then the astronomer proposed to her. She started to cry.* (**Astronomer Proposal**).

There are two possible reasons for the movie star starting to cry after the astronomer proposed to her: either the proposal made her extremely happy, or the proposal made her extremely sad. Perhaps the most likely reason for her crying was that the proposal reminded her of murdered lover, therefore making her upset and sad. REMIND, however, does not have the complex knowledge about memories and how they affect people's emotions that would be

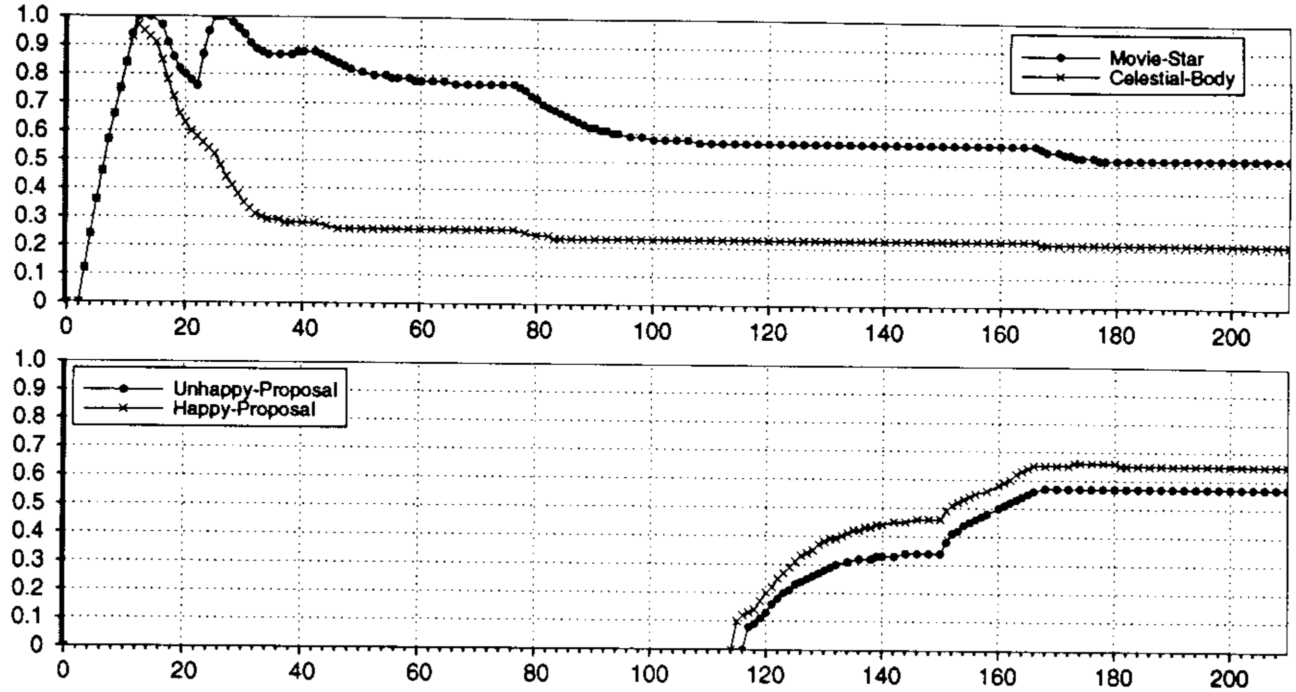

Figure 18. Activations of ambiguous meanings of word *star* and of Happy-Proposal and Unhappy-Proposal interpretations of Propose-Marriage after activation is presented for **Astronomer Proposal**.

necessary to make that interpretation. However, REMIND is given the knowledge that a person will become sad when someone they love dies (Unhappy-Dead-Friend). The network also knows that marriage proposals can be either happy events (Happy-Proposal) or sad events (Unhappy-Proposal), as in ATLAST (Eiselt, 1987).

When the network is presented with input for **Astronomer Proposal**, the word *star* is quickly disambiguated to Movie-Star because of the selectional restrictions that only Humans can be the Actors of Loves (Figure 18). REMIND then infers that the $Shooting causes the plumber to be Dead, and that the Movie-Star that Loves him will therefore be sad (Unhappy-Dead-Friend leading to Unhappy). After input for the phrase *the astronomer proposed to her* is presented, the network infers at about cycle 120 that there are two possible results from this Marriage-Proposal: that she will consider it a Happy-Proposal or an Unhappy-Proposal. These inferences then instantiate the more general Happy and Unhappy frames, respectively, both of which connect to the Movie-Star's Crying, because Crying can be the result-of states Happy or Unhappy.

As shown in Figure 18, Happy-Proposal initially becomes more highly activated than Unhappy-Proposal. This is a result of the network being biased to normally consider Marriage-Proposals to be Happy-Proposals — the weight from Marriage-Proposal to Happy-Proposal is 0.6, but the weight from Marriage-Proposal to Unhappy-Proposal is only 0.4. The gap in Happy-Proposal and Unhappy-Proposals activations begins to narrow at about cycle 150 when both begin to receive feedback from Crying (*she started to cry*). This narrowing occurs because (a) Crying has a higher weight to Unhappy than to Happy and (b) Unhappy is already highly active from Unhappy-Dead-Friend. However, REMIND's bias towards Marriage-Proposals being Happy-Proposals is too great, and Happy-Proposal finishes with more activation than Unhappy-Proposal. The final interpretation of **Astronomer Proposal** is therefore that (1) the Movie-Star was made happy by the astronomers proposal and started to cry tears of joy, and (2) she was also sad because her lover the plumber was killed (an active inference path for which REMIND could find no causal connection to the crying of the marriage proposal).

As **Astronomer Proposal** illustrates, REMIND often comes up with counter-intuitive interpretations of stories when the biases of its connection weights are too strong or when it does not have enough knowledge to make the needed inferences for the right interpretation. However, when there is a highly-analogous episode (or case) in memory, the influence of episodic retrieval upon text understanding can lead REMIND to a correct interpretation of its input. For example, consider:

Suzie loved George, but he died. Then Bill proposed to her. She became sad. **(Sad Proposal)**

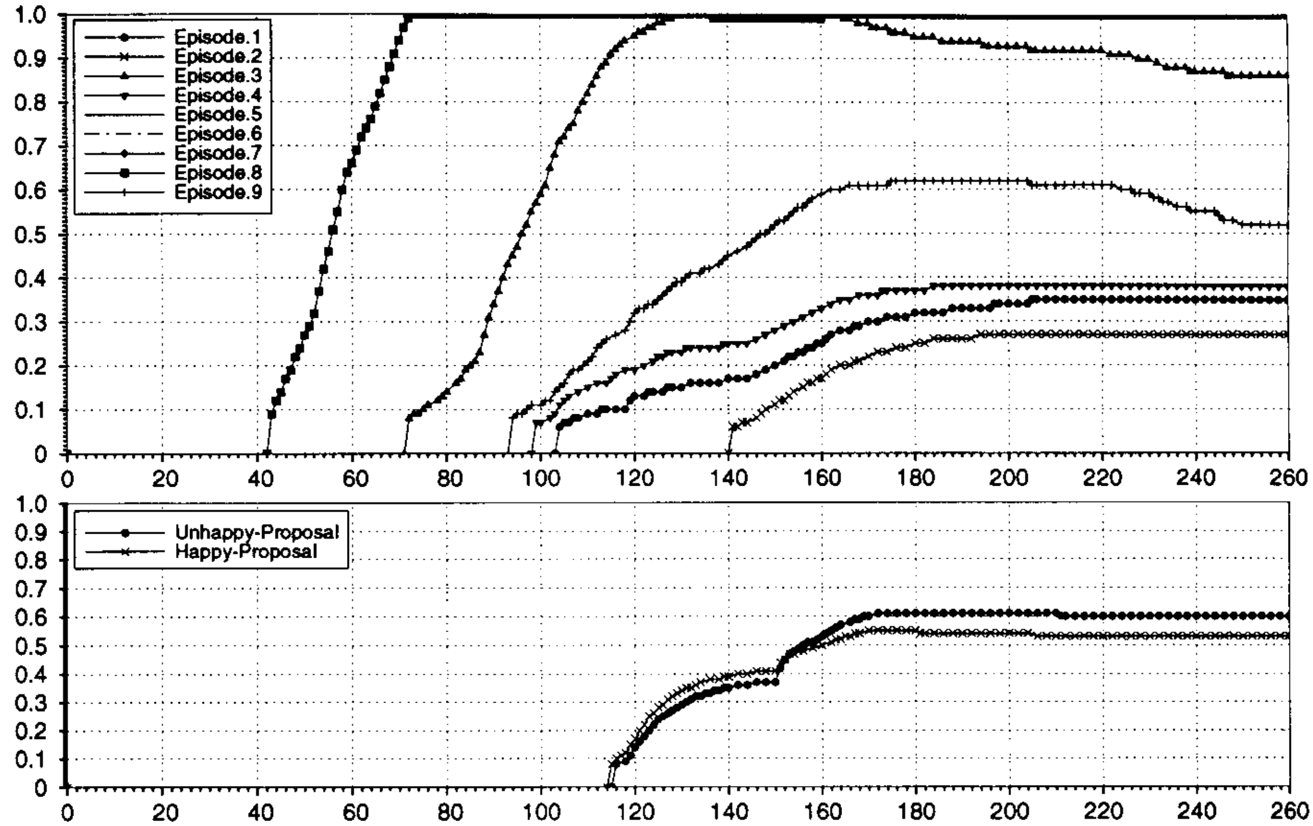

Figure 19. Activations of episodes and of Happy-Proposal and Unhappy-Proposal interpretations of Propose-Marriage after activation is presented for **Astronomer Proposal** in network containing **Sad Proposal** episode.

Sad Proposal is quite similar to **Astronomer Proposal**, except that in **Sad Proposal** the input made it explicit that Suzie became Unhappy after Bill proposed to her. This essentially forces the network to make the correct interpretation, that the Marriage-Proposal after the death of her lover was an Unhappy-Proposal. This interpretation, including the inference Unhappy-Proposal.8, is stored in memory as Episode.8 in Table 3.

Figure 19 shows the activation levels of **Sad Proposal** (Episode.8) and the other episodes as **Astronomer Proposal** is being understood by REMIND. As expected, **Sad Proposal** quickly dominates most of the other episodes because it is so similar to **Astronomer Proposal**. Episode.3 becomes temporarily active because it also involves somebody shooting somebody to death. However, **Sad Proposal** eventually wins and is retrieved.

The most interesting result shown in Figure 19 is the activation levels of the competing Happy-Proposal and Unhappy-Proposal frames. As when **Astronomer Proposal** was presented to the network without any episodes in memory, Happy-Proposal initially has more activation than Unhappy-Proposal because of its higher weight from Propose-Marriage. In this case, however, Episode.8 is highly active, and with it Unhappy.8 and Unhappy-Proposal.8. As described in section 4, these episode instances feed activation back into their concepts in the understanding network. Unhappy-Proposal therefore gets significant activation from Unhappy-Proposal.8. This added evidence allows its activation to climb over that of Happy-Proposal, which gets no added evidence from any of the episodes in memory. When the network settles, Unhappy-Proposal wins over Happy-Proposal, so REMIND's interpretation is that the astronomer's marriage proposal made the movie star unhappy. The network therefore selects the correct interpretation of **Astronomer Proposal** because of activation feedback from an analogous case in memory, **Sad Proposal.**

REMIND's use of the same spreading-activation mechanism for both language understanding and episodic memory retrieval demonstrates one way memory retrieval can subtly affect the interpretation process. When stored episodes share conceptual similarity with a cue that REMIND is comprehending, these episodes feed evidential activation back into the inferencing network. This feedback can bias REMIND's interpretation to be consistent with the active episodes, a limited form of case-based reasoning. These effects emerge entirely from the integration of language understanding and retrieval within a single spreading-activation network.

5.4 Comparison to General Models of Reminding

It is difficult to directly compare REMIND to most case-based reasoning models because they were developed with different goals in mind. As described previously, CBR models are usually models of expert reminding or models built to demonstrate how certain kinds of abstract remindings can occur. Unlike REMIND, they are not meant to be models of general, non-expert human reminding. An advantage of case-based reasoning models over REMIND is that their use of symbolic processing abilities allow them to handle longer and more complex episodes than REMIND (and connectionist models in general) can currently handle. On the other hand, as a model of comprehension and general reminding, REMIND is better able to explain psychological results such as the relatively high prevalence of remindings based on superficial similarities and on how the reminding and language understanding processes interact and effect each other.

The models that REMIND is most directly comparable to are ARCS (Thagard et al., 1990) and MAC/FAC (Gentner & Forbus, 1991), two other simulations of general, non-expert reminding. All three models were built to take into account psychological evidence showing that episodic memory retrieval is strongly influenced by surface feature similarities between a cue and the target episodes in memory, but that deeper analogical or thematic similarities sometimes play an important role. However, there are two important differences between how REMIND explains this evidence compared to ARCS and MAC/FAC.

The most obvious difference is that REMIND is an inferencing-based theory of reminding. Memory retrieval in REMIND results directly from the inferencing and disambiguation process used to understand and form elaborated interpretations of REMIND's cue. ARCS and MAC/FAC, on the other hand, are stand-alone memory retrieval models that are given complete propositional representations of their cues and memory episodes. An advantage of this approach is that it allows ARCS and MAC/FAC to deal with retrieval of much more complicated episodes than does REMIND. ARCS, for example, performed memory retrieval using complex, hand-coded predicate calculate representations of synopses of 24 Shakespearean plays. On the other hand, a major criticism of ARCS and MAC/FAC is that neither model specifies exactly how the representation of its input cues and episodes is formed. More importantly, neither model specifies what kinds of knowledge those representations should generally include. Should the cue representations include only the surface propositions directly stated in phrases of a cue's text? Or should they include a fully elaborated interpretation of the cue, including a complete causal plan/goal analysis of the text and any abstract themes it involves, as in REMIND?

We believe that many types of memory retrieval cannot be performed without such inferences, as illustrated by some of the examples in this paper (e.g., section 4.3 and 5.1) and some of the examples of case-based reasoning systems. However, even if stand-alone retrieval models such as ARCS and MAC/FAC used fully elaborated interpretations of their cues, we believe that not modeling the *process* by which these interpretations are formed misses an important factor in reminding. People read texts with widely different levels of analysis, ranging from simply skimming the text to reading it carefully and thinking deeply about its ramifications. While there are circumstances under which it is reasonable to assume that subjects make relatively deep thematic inferences, it is misleading to think that this is always the case (e.g., Seifert et al., 1986). Thus, when the process by which the retrieval cue is constructed itself is not modeled, there is no way to simulate the specific circumstances under which understanders infer and can use planning or thematic information in probing memory. In contrast, REMIND explicitly models the cue interpretation process, and so can potentially explain when elaborated abstract inferences are available to affect reminding.

Another major difference between REMIND and ARCS and MAC/FAC is in how they theorize that analogical similarity exerts an influence on memory retrieval. Both ARCS and MAC/FAC perform memory retrieval in two stages. In their first stage, both search memory to find the episodes in memory sharing the most surface semantic commonalties with the cue. In their second stage, they compute which of the contacted episodes best match the cue and should be retrieved. In addition to counting surface similarities, they calculate the degree of *structural isomorphism* (or *relational consistency)* as an explicit factor in their computation of which episode best matches the cue. Eskridge (in press) describes a hybrid connectionist model of analogical retrieval and transfer that acts in a similar way.

Isomorphism can best be explained by an example from Thagard et al. (1990) for the cue *The dog bit the boy and the boy ran away from the dog* (**Boy Run**). Compare this to the analogs *Fido bit John and John ran away from Fido* (**John Run**) and *Rover bit Fred and Rover ran away from Fred* (**Rover Run**). **John Run** is structurally isomorphic with **Boy Run**, because mapped objects play the same roles in mapped predicates. In both cases, the dog did the biting and the person it bit did the running. In **Rover Run**, however, it was the dog that ran from the person it bit. **John Run** is therefore more isomorphic, in a purely syntactic sense, to **Boy Run** than is **Rover Run**, and is therefore a better analog.

Both ARCS and MAC/FAC explicitly compute the level of syntactic isomorphism between a cue and episodes that share surface semantic overlap with the cue. The degree of cue/target isomorphic match is factored into their second stages matching process. Analogical similarity is hypothesized by these models to exert its effect on memory retrieval as a direct result of this specifically computed degree of syntactic relational consistency between cues and memory episodes. REMIND, on the other hand, never explicitly computes the degree of isomorphism between a cue and memory episodes. In REMIND, the influence of such relational consistency is entirely the result of the inferencing process. Relationally consistent targets are retrieved over relationally inconsistent targets in REMIND only when the different syntactic structure of each input *leads to different inferences*. For example, if presented with **John Run**, REMIND would infer that the boy ran away because he was afraid that dog would continue its attack. However, if presented with **Rover Run**, REMIND would infer that the dog ran away because it feared retaliation in the form of anger or a kick from the boy. Because of the different inferences and interpretation of the two episodes, REMIND, like ARCS and MAC/FAC, would therefore retrieve **John Run** when presented with **Boy Run** as a cue. Unlike ARCS and MAC/FAC, however, REMIND does so without having to go through a separate stage to explicitly compute the degree of syntactic isomorphism.

Like ARCS, our earlier model of integrated language understanding and memory retrieval, SAARCS (Lange, Melz, Wharton, and Holyoak, 1990), included relational consistency as an explicit factor in reminding. SAARCS was a hybrid connectionist model that combined ROBIN (Lange & Dyer, 1989) with aspects of ARCS. Like REMIND, SAARCS used the ROBIN portion of the network to infer and disambiguate an interpretation of a cue. Unlike REMIND, SAARCS then explicitly calculated relational consistency to build a constraint satisfaction mapping network like ARCS' to determine which episode was retrieved. Oftentimes SAARCS would not have enough knowledge to make different inferences between cues with cross-mapped bindings, such as for *The boat followed the dolphins* vs. *The dolphins followed the boat*. In those cases, SAARCS' use of ARCS' explicit influence of relational consistency between cues and targets led SAARCS to retrieve the right analog (Lange et al., 1990). However, when SAARCS had enough knowledge to build a different interpretation for cross-mapped cues (such as for *The surfer ate the shark* vs. The *shark ate the surfer)*, the explicit effect of relational consistency turned out to be unneeded. As in REMIND, the different shared inferences in SAARCS were enough to activate the analogous episode enough to win. It turned out that in every case that we built enough knowledge into the network for it to build different interpretations for disanalogous episodes, the degree of syntactic isomorphism only boosted (confirmed) the analogous episode that had already won due to those different interpretations. This was a primary motivating factor for simplifying the model and moving to REMIND, a purely inferencing-based model of episodic reminding.

We therefore believe that the noted effects of syntactic isomorphism and relational consistency on memory retrieval can be fully explained by the understanding process. Relationally consistent episodes tend to have similar inferences, interpretations, and themes, while relationally inconsistent episodes tend to have different inferences, interpretations, and themes. In REMIND, relationally-based reminding occurs as a natural side-effect of interpreting and disambiguating an input text. Relational consistency only affects reminding to the degree that it changes those inferences.

5.5 Future Work

In the future, there are three main areas that we would like to explore: (1) improved inferencing abilities, (2) the ability to determine the initial surface role-bindings with additional lexical information in the networks, and (3) automatic learning of the episodes the network has understood.

5.5.1 Improved Inferencing

REMIND's propagation of signature activations dynamically instantiates candidate inference paths in parallel, in much the same way as marker-passing systems. The use of ROBIN's signatures therefore allows REMIND to perform dynamic inferencing difficult for most connectionist models, while using its evidential activation allows it to perform disambiguation and reinterpretation difficult for most symbolic models. However, REMIND's representation and rule-firing abilities are currently limited relative to those of traditional symbolic models, limiting the length and complexity of the texts the model can understand and therefore remember.

One of the main restrictions of the model as described is that there can be only one dynamic instance of each frame at any given time, since binding units can only hold one signature activation at once. Because of this, REMIND cannot yet represent or interpret texts involving two different seeing or eating events, for instance. Another limitation is that REMIND currently only propagates signatures of *pre-existing* concepts, such as of Cooking-Pot, Marijuana, or John. REMIND does not propagate signatures of the *dynamically* instantiated frames inferred by signatures, such as the dynamic instance of Cooking-Pot or Marijuana being Inside-Of a Dishwasher in Figure 5. Not being able to propagate signatures of dynamically instantiated frames makes it impossible for REMIND to encode most rules for *general* planning knowledge or complex interactions of goals, which generally require the ability to reason over any dynamic plan or goal instance the system might have. These type of rules are needed to understand many complex texts, such as those involving abstract planning failures or themes (cf. Schank, 1982; Dyer, 1983).

These and other restrictions on the types of inferencing REMIND performs limit the complexity of the episodes REMIND can currently understand and retrieve relative to many symbolic language understanding and case-based reasoning models. We are currently exploring solutions to some of these problems to allow multiple dynamic instantiations of individual frames and to allow rules that propagate signatures of the novel concepts inferred by the network. A number of different ways to approach these problems and handle more complex text are discussed in Lange (1992). Shastri and Ajjanagadde (in press) discuss an analogous solutions for structured networks that do not perform disambiguation.

5.5.2 Lexical Information and Initial Role-Bindings

REMIND does not currently address the problem of deciding upon the original syntactic bindings, e.g. that *pot* is bound to the Object role of a phrase. Rather, REMIND's networks are given these initial bindings and use them for high-level inferencing. To handle natural language input entered as text, the network must somehow contain and use syntactic and phrasal information to create the initial role-bindings that REMIND is currently given by hand.

5.5.3 Automatic Learning of Episode Units

The representations that REMIND uses for its memory episodes are created entirely by REMIND's spreading-activation understanding process. To store those representations in the network, however, the units and connections used to encode them must be added by hand. It would be desirable to have those episode units be learned automatically by the network itself. We are currently exploring a mechanism to *recruit* units (cf. Diederich, 1990) to encode the interpretation held by signature and evidential activation in the network and therefore allow episode learning to proceed without intervention.

Another area to explore will be gradual decay of the weights between episode units and the semantic network. Episodes that are not retrieved for a long time should gradually have their weights decay so that they become more difficult to become reminded of as time goes on. This is also a potential solution to the problem of indexing too many episodes under general concepts, such as Possess-Obj and Inside-Of, since connections from them to old episodes could gradually decay away and become available for new ones.

6. Conclusions

The process of memory retrieval has generally been explored in isolation from the process of language understanding, even though both activities seem to share many of the same processes. One of the reasons for this is that building an integrated model of language understanding and retrieval requires not only solving the important aspects of each of the processes individually — a difficult enough problem separately — but also finding a parsimonious way to integrate the two processes that can explain the effect they have on each other. Because of their simple processing mechanism and demonstrated abilities to perform disambiguation and model the psychological effects of priming and context, structured spreading-activation networks seem to have a great deal of potential for building such an integrated model.

A number of difficult problems for spreading-activation networks have to be solved to use them for language understanding and memory retrieval. First, language understanding requires the ability to represent variable bindings within the network and to perform dynamic inferencing to explain the plans and goals of actors. Second, problems of disambiguation and frame selection require the ability to combine evidential domain knowledge with the contextual evidence of the network's inferences to select the best interpretation of a text from multiple possible interpretations. In addition, ambiguous texts requiring many inferences make the problems of controlling crosstalk between unrelated concepts and inferences especially important to solve. And finally, a method for representing episodes in the network's long-term memory that allows them to be influenced and retrieved by the spreading-activation understanding process must be developed.

This paper has described REMIND, a structured spreading-activation model that solves many of these problems in an integrated model of language understanding and episodic reminding. In REMIND, activation is spread through a semantic network that performs dynamic inferencing and disambiguation to infer a conceptual representation of an input cue. Because stored episodes are associated with concepts used to understand them, the spreading-activation process also activates any memory episodes in the network that share features or knowledge structures with the cue. After the cue's conceptual representation is formed, the network recalls the memory episode having the highest activation. Since the inferences made from a cue often include actors plans and goals only implied in a cue's text, REMIND is able to get abstract, analogical remindings that would not be possible without an integrated understanding and retrieval model.

Theoretically, REMIND lies somewhere between case-based reasoning models and general analogical retrieval models such as ARCS and MAC/FAC. Like ARCS and MAC/FAC, REMIND is meant to be a psychologically-plausible model of general human reminding, and therefore takes into account the prevalence of superficial feature similarities in remindings. However, we believe that many of the types of high-level planning and thematic knowledge structures used as indices in case-based reasoning systems also have an important effect on reminding. REMIND is thus partially an attempt to bridge the gap between case-based and analogical retrieval models. As it turns out, this gap is naturally bridged when the same spreading-activation mechanism is used to both understand cues and to retrieve episodes from memory. Using the same mechanism for both processes causes retrieval to be affected by all levels that a text was understood with. This is the case in REMIND, in which the understanding mechanism is given the superficial features and actions of a text and attempts to explain them by inferring the plans and goals being used causing long-term memory episodes to be activated by both.

Although being an integrated comprehension and retrieval model makes REMIND more complex than the retrieval mechanisms of case-based reasoning models and ARCS and MAC/FAC, it also allows REMIND to be simpler than them in a couple of significant respects. A large amount of the research in case-based reasoning models is devoted towards discovering the best indices to store cases under (the *indexing problem).* In REMIND, however, episodes are simply stored (indexed) under all of the concepts that played a part in understanding them. When combined with the comprehension part of the model, this simple connectionist approach to indexing avoids the indexing problem altogether, while still giving the effect of having chosen the proper indices since the most salient and unique features of an episode in a given context naturally become highly activated as part of the understanding process.

ARCS and MAC/FAC also avoid the indexing problem of CBR models because they both make contact with episodes that share any feature similarities with the cue, in effect using all features as indices. Where REMIND differs from ARCS and MAC/FAC is that both ARCS and MAC/FAC use separate mechanisms to explicitly factor the degree of syntactic relational consistency (analogical similarity) into retrieval. We believe that psychological effects of analogical similarity on memory retrieval that their separate syntactic mechanisms are meant to model can be fully explained by the understanding process. Relationally consistent episodes tend to have similar inferences, interpretations, and themes, while relationally inconsistent episodes do not. In REMIND, analogically-based reminding therefore occurs as a natural side-effect of understanding an input text, rather than as a result of a separate process that explicitly computes it, as in ARCS and MAC/FAC.

A final aspect to note about REMIND concerns how language understanding and retrieval processes come full circle. The episode retrieved depends crucially on the interpretation of the cue from the spreading-activation network's inferences. Once an episode is retrieved, it in turn primes the activation of the evidential spreading-activation network, perhaps leading to a different disambiguation and therefore interpretation of the next cue. Thus, we believe that REMIND is able to uniquely provide insights which other current reminding models are not able to show. As such, REMIND represents an entirely new class of reminding models.

6.1 Acknowledgments

This research was supported by NSF grant DIR-9024251, Army Research Institute contract MDA 903-89-K-0179, and by a grant from the Keck Foundation. We would like to thank John Barnden, Michael Dyer, Keith Holyoak, Eric Melz, and John Reeves for their helpful comments on the model and on earlier drafts of this paper.

7. References

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*, Philadelphia, PA.

Cottrell, G., & Small, S. (1982). A connectionist scheme for modeling word-sense disambiguation. *Cognition and Brain Theory, 6,* 89-120.

Dolan, C. P., & Smolensky, P. (1989). Tensor product production system: A modular architecture and representation. *Connection Science, 1 (1)*, 53-68.

Dyer, M. G. (1983). *In-depth understanding: A computer model of integrated processing for narrative comprehension.* Cambridge, MA: MIT Press.

Eskridge, T. C. (in press). A hybrid model of continuous analogical reasoning. To appear in J. Barnden & K. Holyoak (Eds.), *Advances in connectionist and neural computation theory, volume II: Analogical connections.* Norwood, NJ: Ablex.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41,* 1-63.

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou and A. Ortony (Eds.), *Similarity and analogical reasoning,* (pp. 199-241). New York, NY: Cambridge University Press.

Gentner, D., & Forbus, K. D. (1991). MAC/FAC: A model of similarity-based retrieval. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 504-509). Hillsdale, NJ: Lawrence Erlbaum.

Gentner, D., & Landers, R. (1985). Analogical reminding: A good match is hard to find. *Proceedings of the International Conference on Systems, Man and Cybernetics* (pp. 607-613). Tucson, AZ.

Gick, M., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology, 12,* 306-355.

Gick, M. L., & McGarry, S. J. (in press). Learning from mistakes: Inducing analogous failures to a source problem produces later successes in analogical transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition.*

Glucksberg, S., Kreuz, R. J., & Rho, S. H. (1986). Context can constrain lexical access: Implications for models of language comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12,* 323-335.

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* (pp. 227-246). Hillsdale, NJ: Lawrence Erlbaum.

Hammond, K. (1989) *Case-based planning.* Boston: Academic Press.

Hendler, J. (1989). Marker-passing over microfeatures: towards a hybrid symbolic/connectionist model. *Cognitive Science, 13,* 79-106.

Hofstadter, D., & Mitchell, M. (in press). The copycat project: A model of mental fluidity and analogy-making. To appear in J. Barnden & K. Holyoak (Eds.), *Advances in connectionist and neural computation theory, volume II: Analogical connections.* Norwood, NJ: Ablex.

Holldobler, S. (1990). A structured connectionist unification algorithm. *Proceedings of the Ninth National Conference on Artificial Intelligence*, Boston, MA.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition, 15*, 332-340.

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.

Kohonen, T. (1984). *Self-organization and associative memory.* New York: Springer Verlag.

Kolodner, J. (1984). *Retrieval and organizational strategies in conceptual memory: A Computer Model.* Hillsdale, NJ: Lawrence Erlbaum.

Kolodner, J., Simpson, R., & Sycara, K. (1985). A process model of case-based reasoning in problem solving. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA.

Lange, T. (1992). Lexical and pragmatic disambiguation and reinterpretation in connectionist networks. *International Journal of Man-Machine Studies, 36*, 191-220.

Lange, T., & Dyer, M. G. (1989). High-level inferencing in a connectionist network. *Connection Science, 1 (2)*, 181-217.

Lange, T., Hodges, J., Fuenmayor, M., & Belyaev, L. (1989): DESCARTES: Development Environment For Simulating Hybrid Connectionist Architectures. In *Proceedings of the Eleventh Annual Meeting of the Cognitive Science Society* (pp. 698-705). Hillsdale, NJ: Lawrence Erlbaum.

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.

Lebowitz (1980). *Generalization and memory in an integrated understanding system.* Ph.D. thesis, Research Report #186, Yale University, Department of Computer Science, New Haven, CT.

Lytinen, S. (1984). *The organization of knowledge in a multi-lingual integrated parser.* Ph.D. thesis, Research Report #340, Yale University, Department of Computer Science, New Haven, CT.

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. (in press). *Towards subsymbolic natural language processing: An integrated model of scripts, lexicon, and memory.* Cambridge, MA: The MIT Press.

Miikkulainen, R. & Dyer, M. G. (1991). Natural language processing with modular PDP networks and distributed lexicon. *Cognitive Science, 15*, 343-399.

Norvig, P. (1989). Marker passing as a weak method for text inferencing. *Cognitive Science, 13*, 569-620.

Owens, C. (1989). Integrating feature extraction and memory search. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society,* Ann Arbor, MI.

Ratterman, M. J., & Gentner, D. (1987). Analogy and similarity: Determinants of accessibility and inferential soundness. *Proceedings of the Ninth Annual Meeting of the Cognitive Science Society* (pp. 22-34). Hillsdale, NJ: Lawrence Erlbaum.

Reeves, J. F. (1991). *Computational morality: A process model of belief conflict and resolution for story understanding.* Ph.D. thesis, Computer Science Department, University of California, Los Angeles.

Riesbeck, C. K., & Schank, R. C. (1989). *Inside case-based reasoning.* Hillsdale, NJ: Lawrence Erlbaum.

Riesbeck, C. K., & Martin, C. E. (1986). Direct memory access parsing. In J. Kolodner & C. Riesbeck (Eds.), *Experience, Memory, and Reasoning*, pp. 209-226. Hillsdale, NJ: Lawrence Erlbaum.

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): A General Framework for Parallel Distributed Processing. In Rumelhart & McClelland (eds.), *Parallel Distributed Processing: Vol. 1*, p. 45-76. Cambridge, MA: The MIT Press.

Schank, R. C. (1982). *Dynamic memory.* New York: Cambridge University Press.

Schank, R. C., & Abelson, R. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Lawrence Erlbaum.

Schank, R. C., & Leake, D. B. (1989). Creativity and learning in a case-based explainer. *Artificial Intelligence, 40,* 353-385.

Seifert, C. M. (in press). The role of goals in retrieving cases. To appear in J. Barnden & K. Holyoak (Eds.), *Advances in connectionist and neural computation theory, volume II: Analogical connections.* Norwood, NJ: Ablex.

Seifert, C. M., McKoon, G., Abelson, R. P., & Ratcliff, R. (1986). Memory connections between thematically similar episodes. *Journal of Experimental Psychology: Human Learning and Memory, 12,* 220-231.

Shastri, L., & Ajjanagadde, V. (in press). From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. To appear in *Behavioral and Brain Sciences.*

St. John, M. (in press). The story gestalt: A model of knowledge intensive processes in text comprehension. *Cognitive Science.*

Sun, R. (in press). Beyond associative memories: Logics and variables in connectionist models. To appear in *Information Sciences.*

Swinney, D. A. (1979). Lexical access during sentence comprehension: (Re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior, 15,* 681-689.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46,* 259-310.

Till, R. E., Mross, E. F., & Kintsch, W. (1988). Time course of priming for associate and inference words in a discourse context. *Memory & Cognition, 16,* 283-298.

Touretzky, D. (1990). Connectionism and compositional semantics. In J. A. Barnden & J. B. Pollack (Eds.), *Advances in connectionist and neural computation theory,* Norwood, NJ: Ablex.

Touretzky, D., & Hinton, G. (1988). A distributed connectionist production system. *Cognitive Science, 12,* 423-466.

Waltz, D., & Pollack, J. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science, 9,* 51-74.

Wilensky (1983). *Planning and Understanding.* Reading, MA: Addison-Wesley.

Wharton, C. M, Holyoak, K. J., Downing, P. E., Lange, T. E., & Wickens, T. D. (1991). Retrieval competition in memory for analogies. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 528-533). Hillsdale, NJ: Lawrence Erlbaum.

Wharton, C. M, Holyoak, K. J., Downing, P. E., Lange, T. E., & Wickens, T. D. (1992). The story with reminding: Memory retrieval is influenced by analogical similarity. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.