# Lexical and Pragmatic Disambiguation and Reinterpretation in Connectionist Networks†*

Trent E. Lange

Artificial Intelligence Laboratory
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90024
lange@cs.ucla.edu

**ABSTRACT**

Lexical and pragmatic ambiguity is a major source of uncertainty in natural language understanding. Symbolic models can make high-level inferences necessary for understanding text, but handle ambiguity poorly, especially when later context requires a reinterpretation of the input. Structured connectionist networks, on the other hand, can use their graded levels of activation to perform lexical disambiguation, but have trouble performing the variable bindings and inferencing necessary for language understanding. We have previously described a structured connectionist model, ROBIN, which overcomes many of these problems and allows the massively-parallel application of a large class of general knowledge rules. This paper describes how ROBIN uses these abilities and the contextual evidence from its semantic networks to disambiguate words and infer the most plausible plan/goal analysis of the input, while using the same mechanism to smoothly reinterpret the input if later context makes an alternative interpretation more likely. We present several experiments illustrating these abilities and comparing them to those of other connectionist models, and discuss several directions in which we are extending the model.

## 1. INTRODUCTION

Ambiguity is a major cause of uncertainty in natural language understanding. One aspect of the ambiguity problem is that of resolving *lexical* ambiguity to decide upon the meaning of a word meant in a given phrase. Nearly all of the most frequently used words in English are highly ambiguous. The word *go*, for example, has 63 meanings in the Merriam Webster Pocket Dictionary [Hirst, 1983]. An additional problem is that of resolving *causal* (or *pragmatic*) ambiguity to infer the most plausible reason for some action out of many possible explanations. As an example of both of these problems, consider the phrase:

**P1**: *John put the pot inside the dishwasher*

---

†Appears in *International Journal of Man-Machine Studies,* **36**, 191-220, 1992.

*This research has been supported by grants from the W.M. Keck and ITA Foundations. The author would like to thank Michael Dyer, Keith Holyoak, Risto Miikkulainen, and Charles Wharton for valuable comments and discussions on the contents of this paper.

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

**P2**: *because the police were coming.*

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

While there have been a wide variety of approaches to natural language understanding and the problems of disambiguation and reinterpretation, there have been few attempts to address all three problems in a single model. Symbolic models have had a degree of success in performing the high-level inferencing necessary to understand text, but are relatively brittle in the face of ambiguity and the need for reinterpretation. Connectionist models, on the other hand, have been better suited to resolving the multiple constraints from context often necessary for disambiguation, but have been limited as models of natural language understanding because of their difficulties with performing variable bindings and inferencing.

We have previously described a structured connectionist model, ROBIN (ROle Binding and Inferencing Network), which overcomes some of the problems of connectionist models by propagating activation patterns serving as concepts' *signatures* to allow variable binding and the massively-parallel application of a large class of general knowledge rules [Lange & Dyer, 1989]. Because the structure providing these abilities is embedded within a normal semantic spreading-activation network, ROBIN is able to disambiguate and reinterpret input for sentences that require a number of inferences to understand.

This paper describes how ROBIN uses the contextual evidence from its semantic networks, in combination with its inferencing abilities, to perform lexical and pragmatic disambiguation and to smoothly reinterpret the input if later context makes an alternative interpretation more likely[1]. Several experiments illustrating these abilities and comparing them to those of other connectionist models are presented, along with a discussion of several directions in which we are extending the model.

## 2. PREVIOUS APPROACHES

There are four major types of computational models dealing with disambiguation and/or reinterpretation for natural language understanding. Symbolic rule-based models have concentrated mainly on the abilities needed for understanding complex language, but often have specialized rules for disambiguation. Symbolic marker-passing models also have the basic inferencing abilities necessary for language understanding, but have had more success than rule-based models for disambiguation and reinterpretation due to their massively-parallel generation of candidate interpretations. On the other end of the spectrum, distributed connectionist models are able to resolve ambiguities based on contextual constraints from the surface semantics of the input and biases learned from a training set. Finally, structured (or localist) spreading-activation models can also resolve ambiguities based on surface context, but have the potential to hold and consider multiple interpretations in parallel.

---

[1]ROBIN does not currently address syntax or syntactic ambiguity.

## 2.1. Symbolic Rule-Based Systems

Symbolic rule-based systems have so far had the most success performing the high-level inferencing necessary for natural language understanding. A good example is BORIS [Dyer, 1983], a program for modelling in-depth understanding of relatively long and complex stories. BORIS had a symbolic knowledge base containing knowledge structures representing various actions, plans, goals, emotional affects, and methods for avoiding planning failures. When reading in a story, BORIS would fire rules from its knowledge base to perform inferencing and form an internal representation of the story, about which it could then answer questions. Other models that have successfully approached complex parts of the language understanding process have all had similar types of knowledge representation and rule-firing capabilities.

In order to perform disambiguation, expectation-based conceptual analyzers such as CA [Riesbeck, 1975] and BORIS [Dyer, 1983] use bottom-up or top-down *requests* or *demons* that are activated when 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 thrown away. This makes it impossible to reinterpret the word with new context (as in **Hiding Pot**). A way to get around this problem is to keep the word around in case new context causes a new disambiguation request to fire, such as a request that interprets *pot* as Marijuana in the context of Police. Unfortunately, there is no clean-cut way to decide between conflicting disambiguation rules. For example, one cannot simply specify that the request involving the Police context always has a higher priority than the request involving the Cleaning context, since 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.

Because of these kinds of problems, few rule-based systems go beyond disambiguation to attempt reinterpretation. One exception is Granger's ARTHUR [1980], which supplants incorrect inferences by maintaining a map of pointers to all inferences generated during the processing of text, even if they do not appear in the final representation. While this allows ARTHUR to perform reinterpretation, it is still faced with the problem of how to resolve disambiguation conflicts. It also has the problem that its rule application is fundamentally serial, as in most rule-based systems, slowing it down dramatically as the number of inferencing and disambiguation rules increases.

## 2.2. Marker-Passing Systems

Marker-passing models operate by spreading symbolic markers in parallel across labelled semantic networks that hold their world knowledge. Individual *nodes* represent concepts, and *connections* (or *links*) between those nodes represent the relationships between those concepts. Interpretation of the input is achieved when propagation of markers finds a path of nodes connecting words and concepts from the input text. Because of the symbolic information held in their markers and networks, they have also been able to perform high-level inferencing for natural language understanding (cf. [Quillian, 1969], [Charniak, 1986], [Riesbeck & Martin, 1986], [Granger *et al.*, 1986], and [Norvig, 1989]).

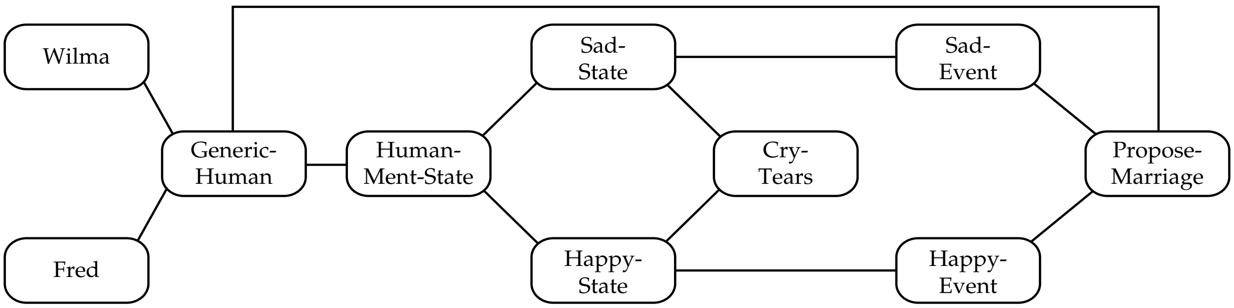


**Figure 1.** Marker-passing network from [Eiselt, 1987].

Lexical and pragmatic ambiguities make themselves known to marker-passing models when multiple candidate paths are returned by the spread of markers. To resolve these ambiguities, they generally use a serial heuristic path evaluator separate from the marker-passing process to select the most relevant path from the many paths generated. Such path evaluators usually include rules that select shorter paths over longer ones, reject paths that do not include as much of the input as competing ones, and so forth. As an example, consider the following text (from [Eiselt, 1987]):

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

Interpretating this text requires that a causal relationship be inferred between Fred's proposal and Wilma's crying. One possible reason for her crying was that she was happy about his proposal and crying "tears of joy". Another possible reason is that she was crying because she was saddened or upset by it. To understand this sentence and resolve the ambiguity, ATLAST [Eiselt, 1987] uses the network shown in Figure 1 by passing markers starting from the nodes for Cry-Tears and Propose-Marriage. The first path found by the propagation of markers is Cry-Tears ↔ Happy-State ↔ Happy-Event ↔ Propose-Marriage, returning the "tears of joy" interpretation. A second path found is Cry-Tears ↔ Sad-State ↔ Sad-Event ↔ Propose-Marriage. To disambiguate between these two paths, ATLAST applies an evaluation metric between two competing paths of equal length that selects the oldest path. The Happy-State path was discovered first, and thus remains as the interpretation of the input.

As in rule-based models, most marker-passing models have not addressed the problem of reinterpretation. Eiselt's ATLAST is an exception. To allow reinterpretation, ATLAST stores the inference paths that its path evaluators originally reject from the interpretation of the input text. If any retained path is rediscovered by the marker-passing process, then it and any of its subpaths or superpaths are re-evaluated against competing paths in the current interpretation. Consider what occurs when **Marriage-a** is followed by:

*Wilma was saddened by the proposal.* (**Marriage-b**)

In this case, it turns out that the original inference that Fred's proposal made Wilma happy was incorrect. In the process of spreading markers for **Marriage-b**, ATLAST rediscovers the Sad-State path to Propose-Marriage that was rejected but stored away. The Sad-State path is then re-evaluated against the Happy-State path of the original interpretation. This time it wins because of an evaluation metric that prefers the path that explains the most of the input, completing the reinterpretation.

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

## 2.3. Distributed Connectionist Models

Distributed connectionist models use massively parallel networks of simple processing elements which represent knowledge as patterns of activation across a number of nodes. Distributed connectionist models have many desirable properties, such as their learning rules (that allow stochastic category generalization), their ability to perform noise-resistant associative retrieval, and their robustness to damage.

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

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

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

### 2.4. Structured Spreading-Activation Models

Structured spreading-activation models (cf. [Waltz & Pollack, 1985] and [Cottrell, 1988]) are connectionist models that spread the representation of knowledge over a large number of nodes, rather than concentrating their knowledge in the weights between a single input and output layer, as most distributed connectionist models do. Structured models represent knowledge in semantic networks similar to those of marker-passing networks, except that nodes are instead simple numeric processing elements and that connections between nodes have weights representing their strengths. The numeric activation level on each conceptual node represents the amount of *evidence* available for its concept in a given context. As in marker-passing networks, structured networks have the potential to pursue multiple candidate interpretations of a story in parallel as each interpretation is represented by activation in different local areas of the network.

Spreading-activation networks are ideally suited for lexical disambiguation because it is achieved automatically as related concepts under consideration provide *graded* activation evidence and feedback to one another in a form of analog constraint relaxation. The weights between nodes are generally based upon the frequency with which two concepts are correlated, therefore allowing certain paths to be biased over others. For example, the average shipyard worker would more likely think of a harbor when hearing the word *port* than of a dark red wine from Portugal. A structured spreading-activation network would model this by having a stronger weight to the Harbor node from the *port* node than to the Red-Wine node. This is in contrast to the binary connections of marker-passing networks, in which a path either exists between two nodes or it does not[2].

As an example of how structured spreading-activation models perform lexical disambiguation, consider the sentence:

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

The word *star* could be easily disambiguated to Movie-Star by a 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 2 shows an extended version of the semantic portion of the structured network Waltz & Pollack [1985] built to process **Star-Marriage** and illustrate this effect. After the input nodes for **Star-Marriage** are clamped, the Celestial-Body interpretation of *star* initially acquires more activation than the Movie-Star interpretation because of priming from Astronomer through Astronomy (Figure 3). However, Movie-Star eventually wins out because activation feedback

[2]Two exceptions are the hybrid marker-passing models of Hendler [1989] and Kitano *et al.* [1989], which include some of the features of spreading-activation networks.

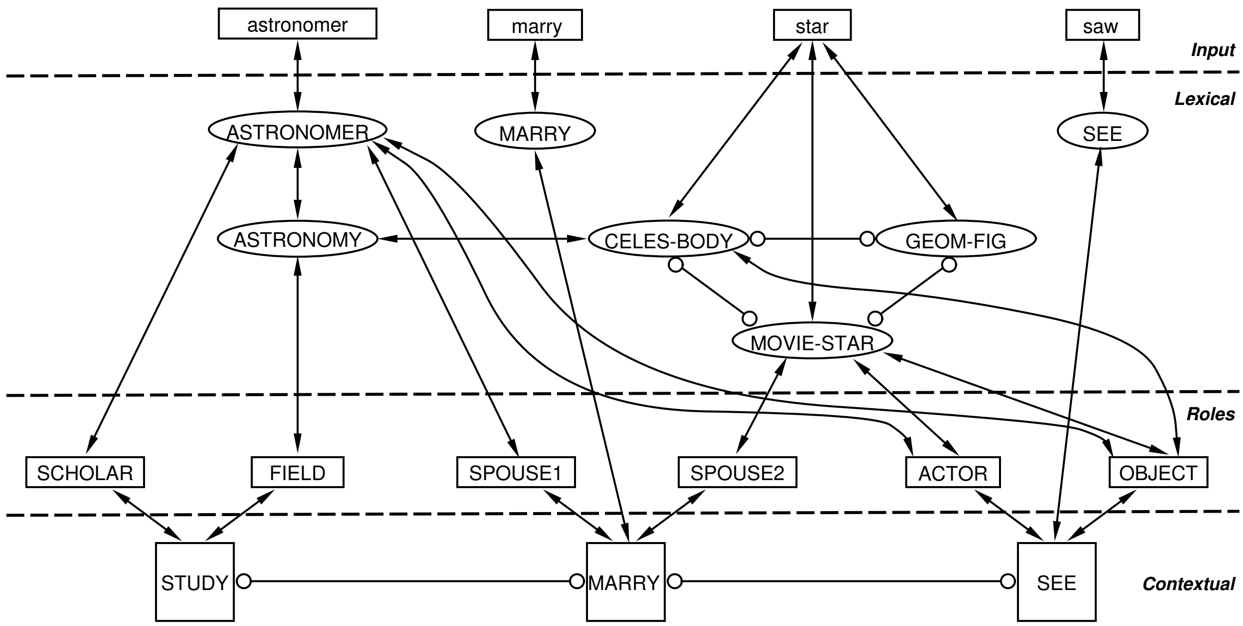


**Figure 2**. Structured spreading-activation network based on [Waltz & Pollack, 1985]. Lines with arrows are excitatory connections; lines with open circles are inhibitory.

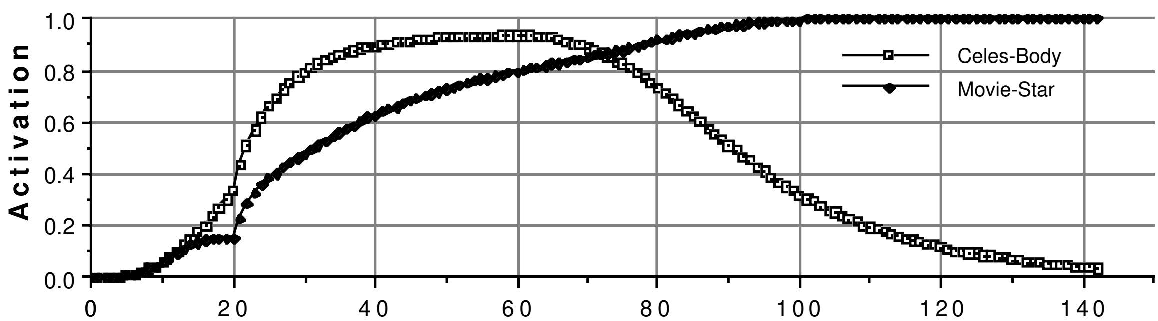


**Figure 3.** Activations of meaning of word *star* after *astronomer married star* is clamped for network in Figure 2.

over the semantic connections from the Marry node to Movie-Star outweighs that spreading from the Astronomer node to Celestial-Body.

Unfortunately, the applicability of spreading-activation models to natural language understanding has been severely hampered because of their inability to represent dynamic role-bindings and perform inferencing[3]. Their lack of variable binding abilities leaves them prone to crosstalk even for simple sentences. For example, the network of Figure 2 has no way to distinguish between the sentences *The astronomer*

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

*saw the star* and *The star saw the astronomer*, despite the crucial difference that the role-bindings make in their interpretation. More importantly, without a mechanism to represent such dynamic bindings, they cannot propagate them as marker-passing systems do to make the chains of inferences necessary for understanding more complex language. This has so far stopped them from going beyond simple disambiguation based on the surface semantic of the input.

## 3. OVERVIEW OF ROBIN

ROBIN [Lange & Dyer, 1989] is a structured spreading-activation model that is able to represent variable bindings and perform some of the general knowledge rules necessary for high-level inferencing, thus partially resolving two of the main weaknesses of structured connectionist networks. Because of this, ROBIN is well-suited to performing lexical and pragmatic disambiguation and allowing reinterpretation. This section gives a short overview of ROBIN and how it performs inferencing, but [Lange & Dyer, 1989] provides a detailed description.

ROBIN uses structured networks of simple connectionist nodes [Feldman & Ballard, 1982] 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. Every frame is related to one or more other frames, with pathways between corresponding roles (representing general knowledge rules) for inferencing.

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

### 3.1. Variable Binding With *Signatures*

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

### 3.2. Propagation of Signatures for Inferencing

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

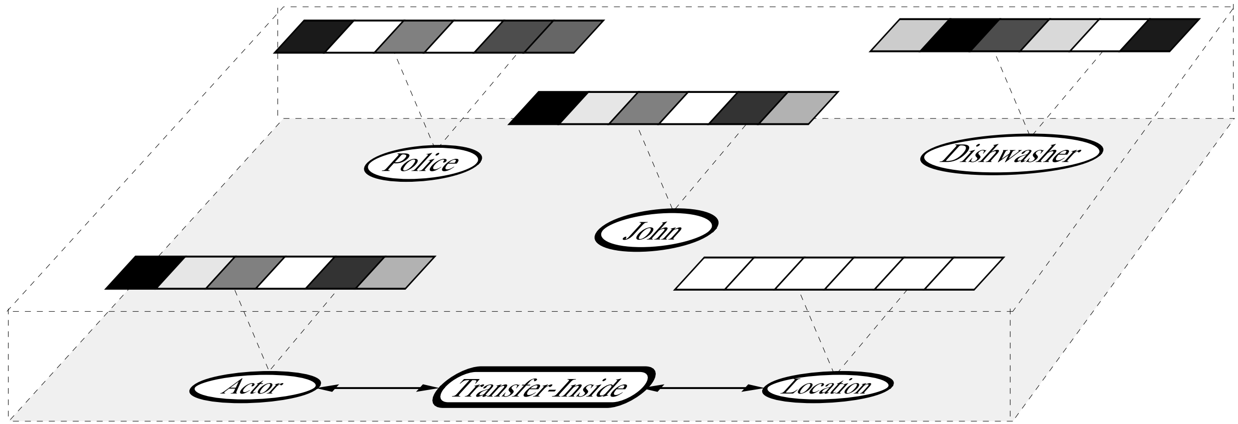


**Figure 4.** Several concepts (ovals on lower plan) and their uniquely-identifying signature patterns, along with the Actor and Location roles of the Transfer-Inside frame. Here each signature is a bank of six units, with increasing levels of activation represented by increasing darkness of shading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

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

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

Figures 5a and 5b illustrate how the network's structure automatically propagates signatures to fire rules such as R1. For simplicity, the signatures in the figure are uniquely-identifying scalar values. Evidential activation for disambiguation is spread through the paths between conceptual nodes on the bottom plane (i.e. Transfer-Inside and its Object role), while signature activation for dynamic role-bindings is spread across the parallel paths of corresponding binding nodes (solid black circles) on the top plane. Nodes and connections for the Actor, Planner, and Location roles are not shown. Initially there is no activation on any of the conceptual or binding nodes in the network.

When input for *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, directly providing activation for concepts John, Transfer-Inside, Cooking-Pot, Marijuana, and Dishwasher. To represent the role-bindings given by phrase **P1**, the binding nodes of each of Transfer-Inside's roles are clamped to the signatures of the concepts bound to them[4]. For example, the binding nodes of Transfer-Inside's Object are clamped to the activations (6.8 and 9.2) of the signatures for objects Marijuana and Cooking-Pot, representing the candidate bindings from the word *pot* (Figure 5a)[5].

---

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

[5]An alternative input, such as *"John put the cake inside the oven",* would be done simply by clamping the signatures of its bindings instead. A completely different set of inferences would then ensue.

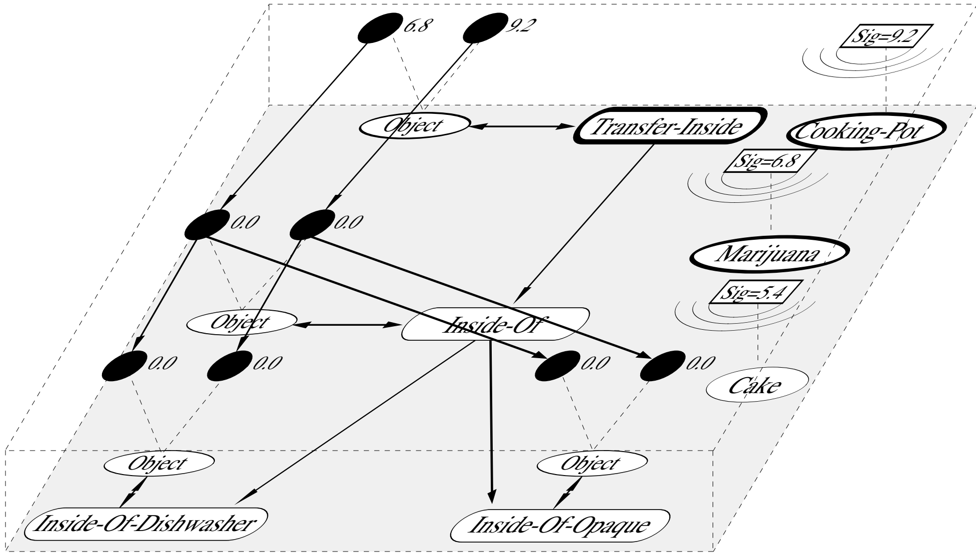


**Figure 5a.** Initial activation for **P1**.

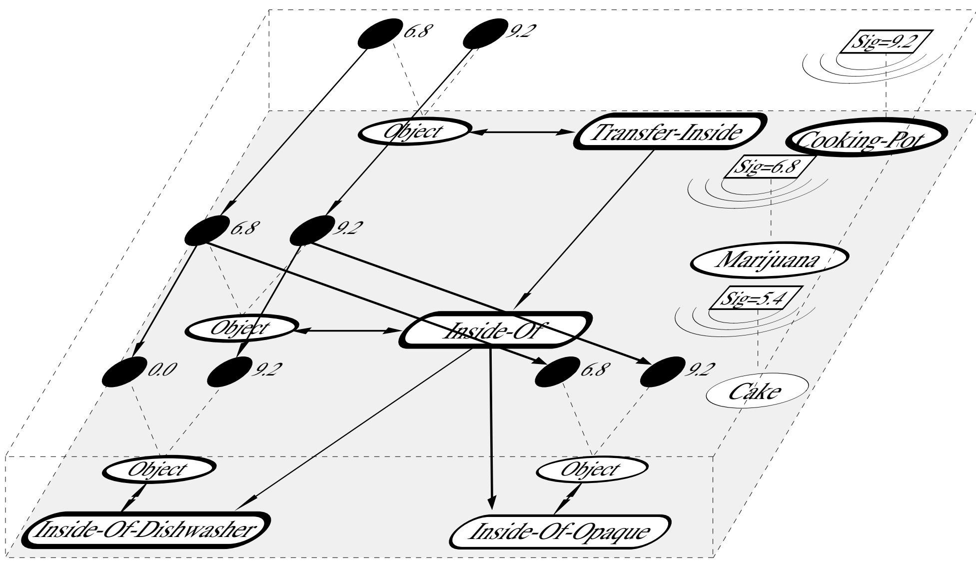


**Figure 5b.** Activation after quiescence has been reached in processing for **P1**.

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

The activation of the network's conceptual nodes 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. The activation of the binding nodes, however, is equal to the maximum of their unit-weighted inputs, allowing signatures to be propagated without alteration.

As activation starts to spread after the initial clamped activation values in Figure 5a, Inside-Of receives evidential activation from Transfer-Inside, representing the strong evidence that something is now inside of something else. Concurrently, the signature activations on the binding nodes of Transfer-Inside's Object propagate to the corresponding binding nodes of Inside-Of's Object (Figure 5b), since each of the binding nodes calculates its activation as the maximum of its inputs. The network has thus made the crucial inference of exactly which thing is inside of the other, by propagating signatures across binding paths encoding rule R1. Similarly, as time goes on, Inside-Of-Dishwasher (representing a kitchen utensil being inside of a dishwasher, a precondition for cleaning) and Inside-Of-Opaque (representing an object being inside of an opaque object, which blocks it from sight) receive evidential activation, with inferencing continuing by the propagation of signature activation to their corresponding binding nodes (Figure 5b).

As can be seen in the figure, propagation of signature activations dynamically instantiates candidate inference paths in parallel in much the same way that marker-passing systems do. As will be seen in the next section, however, the concurrent spread of normal evidential activation along the bottom plane is crucial to ROBIN's abilities to perform disambiguation and reinterpretation without resorting to a separate rule-based path evaluator. One further thing to note is that embedded within the network are nodes that compute and enforce *selectional restrictions* on role-fillers to control the spread of activation when roles' binding constraints are violated. For example, gated links for the selectional restrictions on the Object role of Inside-Of-Dishwasher in Figure 5b do not allow the signature for Marijuana (6.8) to go through, because Marijuana is not cleaned in a dishwasher. In other cases, entire frames remain uninstantiated because their role-fillers' constraints are violated (e.g. Inside-Of-Stove and Inside-Of-Restaurant for phrase **P1**). These selection restrictions (or *logical binding constraints*) dramatically reduce the number of spurious inference paths generated by the propagation of signatures and thus eliminate a large potential source of crosstalk. The network structure imposing selectional restrictions and more details about the rest of ROBIN are described in [Lange & Dyer, 1989].

## 4. DISAMBIGUATION AND REINTERPRETATION IN ROBIN

With the ability to maintain variable bindings and propagate them through the network, ROBIN can now start to approach disambiguation and reinterpretation of language that requires high-level inferencing to understand. The key to ROBIN's disambiguation and reinterpretation abilities is that the propagation of signatures occurs in parallel with the spread of evidential activation along the semantic network of conceptual nodes. The conceptual frame nodes along each alternative inference path have their own levels of evidential activation, influenced by connections from related inferences. The winning path — and thus disambiguated interpretation — is simply the path of nodes with the highest levels of evidential activation. Reinterpretation occurs automatically if new context provides enough activation evidence for another inference path to become the most highly-activated.

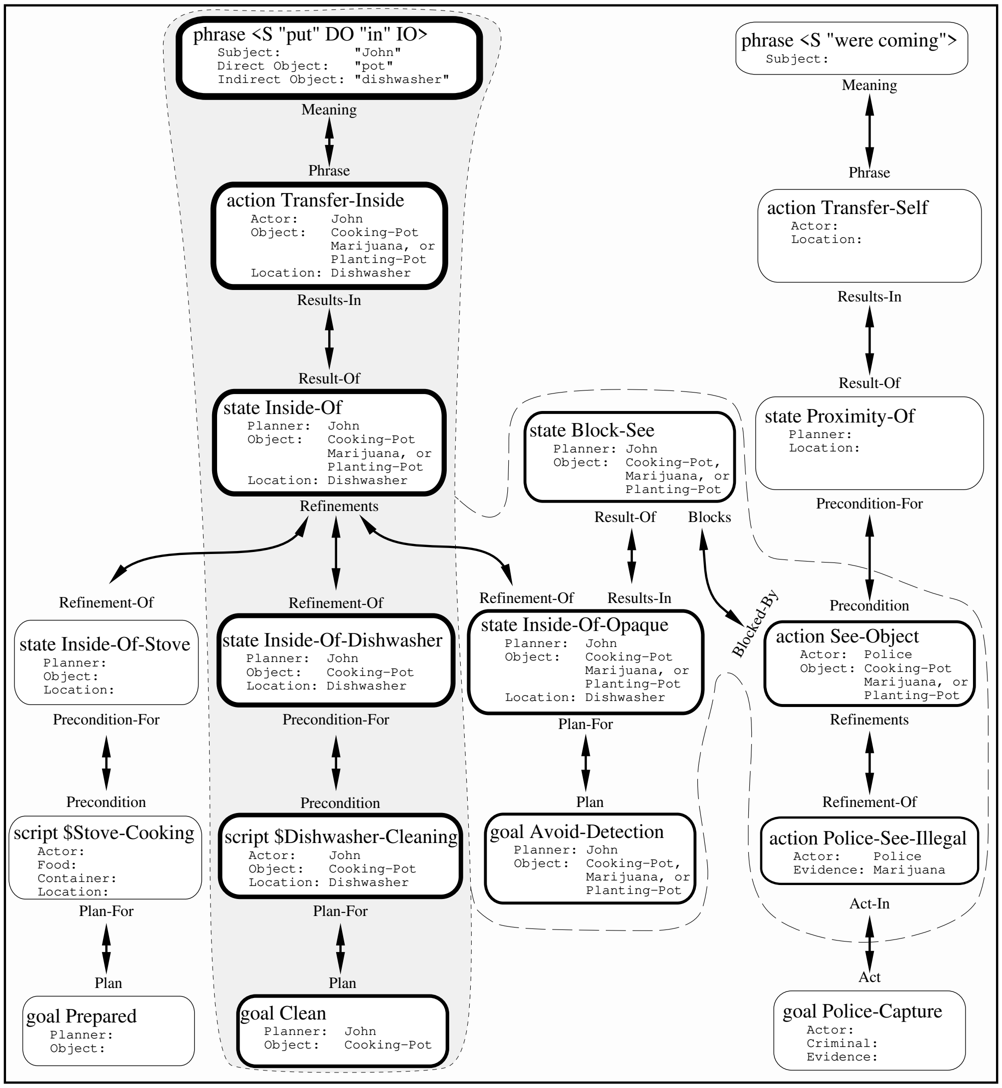


**Figure 6.** Overview of a small portion of a ROBIN semantic network (actually embedded in network structure such as in Figures 5a and 5b) showing inferences dynamically made after clamping of the inputs for phrase **P1**. Thickness of frame boundaries shows the amount of *evidential* activation on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

### 4.1. Disambiguation Example

As an example, consider the disambiguation of phrase **P1** (*John put the pot inside the dishwasher*). Figure 6 shows an overview of the levels of activation and inferences made in a portion of the network after inputs for **P1** have been clamped and activation has settled. As in Figure 5b, the role-bindings of Inside-

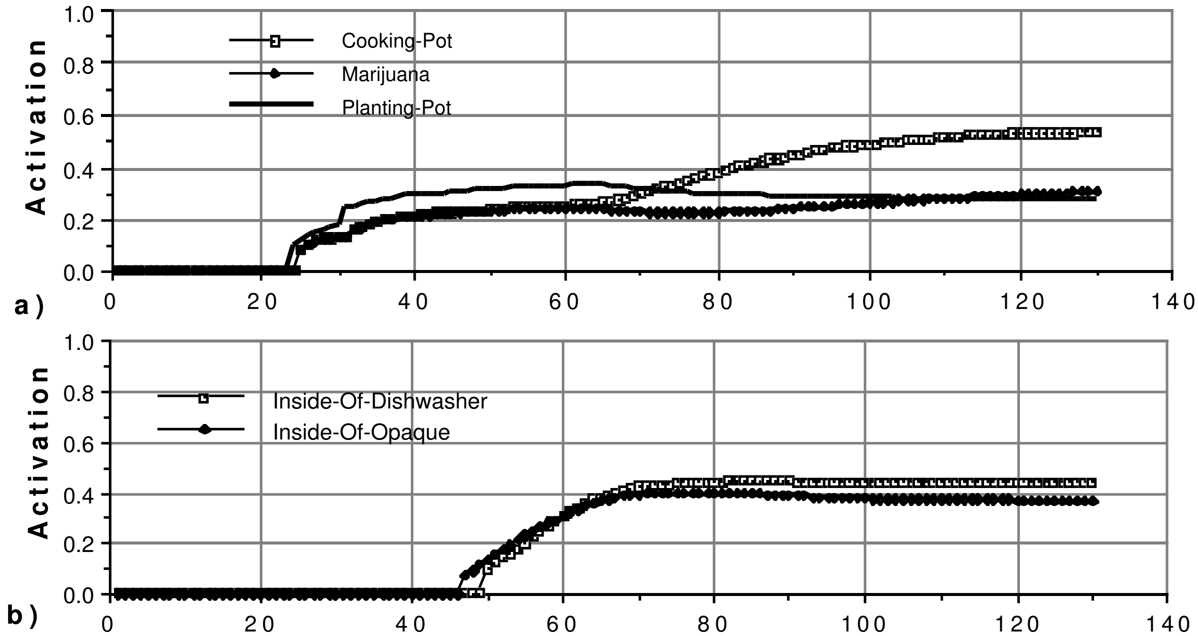


**Figure 7.** Activations of meanings of word *pot* and competing refinements of Inside-Of after clamping for *John put the pot inside the dishwasher* (**P1**).

Of-Dishwasher and Inside-Of-Opaque have been inferred by propagation of signatures. Those activations and evidential activation on the conceptual nodes continued to propagate along the chain of related concepts down to instantiate the Clean goal, and along the other path to goal Avoid-Detection, state Block-See, and finally action See-Object and its refinement Police-See-Illegal, where the level of evidential activation drops below threshold and so stops propagating. Inside-Of-Stove and other refinements of Inside-Of did not receive any activation because a pot inside a dishwasher violated their selectional restrictions.

There are two different possible interpretations of **P1** in the network: the chain running through Inside-Of-Dishwasher to Clean (representing that John was trying to clean a Cooking-Pot), and the chain running through Inside-Of-Opaque to Avoid-Detection, Block-See, and See-Object (representing that John was trying to hide either a Cooking-Pot, Marijuana, or a Planting-Pot)[6]. However, after the spread of activation, Inside-Of-Dishwasher has more evidential activation than Inside-Of-Opaque because of feedback between it and Cooking-Pot and Dishwasher, which have strong connections because of their highly stereotypical usage in $Dishwasher-Cleaning (see plot of activations in Figure 7). The Inside-Of-Dishwasher path is thus chosen as the refinement of Inside-Of, so it and the rest of the frames along the Clean path are chosen as the interpretation of the phrase (represented by the darkly shaded area in Figure 6). Also, because of reinforcement and feedback from the Inside-Of-Dishwasher path, Cooking-

[6]In this network, Avoid-Detection and Block-See are complementary parts of the same chain, since Inside-Of-Opaque is a Plan-For Avoid-Detection and it Results-In Block-See. However, Inside-Of-Dishwasher and Inside-Of-Opaque form different chains, since they are mutually exclusive refinements of Inside-Of.

Pot ended up with more evidential activation than either Marijuana or Planting-Pot (Figure 7). Cooking-Pot is thus selected over the Marijuana and Planting-Pot *bindings* throughout the network.

### 4.2. Structure of the Evidential Network

The disambiguation abilities of ROBIN and other structured spreading-activation networks stem from the fact that their connections naturally combine evidence from context to activate concepts that are most related to the input. This is the case in ROBIN, in which the structure of the conceptual layer causes the activation of each conceptual node to be approximately proportional to the amount of evidence available for it in a given context. Those concepts for which there is the most evidence therefore gain the most evidential activation, and thus become the interpretation of the input.

The evidence combination of structured networks can be understood more clearly if their connectivity is viewed as embedding "heuristics" for activating and selecting interpretations of the input. In general, there is evidence for the selection of a frame if (1) a related frame is active, (2) its roles and fillers are active, or (3) it is the filler of an active frame's role. For example, there is strong evidence that something is Inside-Of something if a Transfer-Inside has happened (1). If Dishwasher is active, then there is a fair likelihood that an Inside-Of-Dishwasher and a $Dishwasher-Cleaning have occurred or will occur, since Dishwasher is the prototypical filler of their Location roles (2). Conversely, if a $Dishwasher-Cleaning is happening, then there is substantial evidence that a Dishwasher is being used (3).

The connectivity of Cottrell & Small [1983] and Waltz & Pollack's [1985] structured networks can be seen to take advantage of the above "selection evidence heuristics" by their direct weighted connections between related concepts, roles, and their prototypical fillers (e.g. Figure 2). However, when networks become larger to incorporate more world knowledge, as in ROBIN, simply having frames receive evidential activation through direct connections from their related frames would cause serious problems. Candidate frames that have potential relations to a large number of frames would always win out over candidates that have a smaller number of related frames. The activation of Inside-Of, for example, would always dominate Transfer-Inside, simply because Inside-Of has a very large number of potential refinement frames. In reality, however, those refinements are mutually exclusive, and only one will be chosen as *the* refinement of a given instantiation of Inside-Of. Thus, the only refinement relation that actually provides evidence for Inside-Of at a given cycle is the one that is most active.

Because of this, connections from related frames pass through an *input branch* node for their relation before they are received by the candidate frame. This is shown in Figure 8, which displays the connections between nodes on a portion of the evidential network centering around frame Inside-Of. For example, the weighted connections from Inside-Of-Dishwasher, Inside-Of-Opaque, and Inside-Of-Stove go into Inside-Of's refinements branch rather than directly into Inside-Of. Relation input branches calculate their activation as the maximum of their inputs — so that only the currently selected (i.e. maximally activated) interpretation provides evidence for the frame[7].

Role nodes, like frame nodes, have several input branches. A role, however, gets evidence only from its competing prototypical fillers (the Prototypes branch) and from the frames that it is used in (the Used-In branch). Each branch calculates its activation as the maximum of its inputs, like the relation input

[7]Input branches are analogous to the input *sites* on "case" units in [Cottrell & Small, 1982], which caused each case unit to receive activation only from its maximally-activated prototypical filler and predicate.

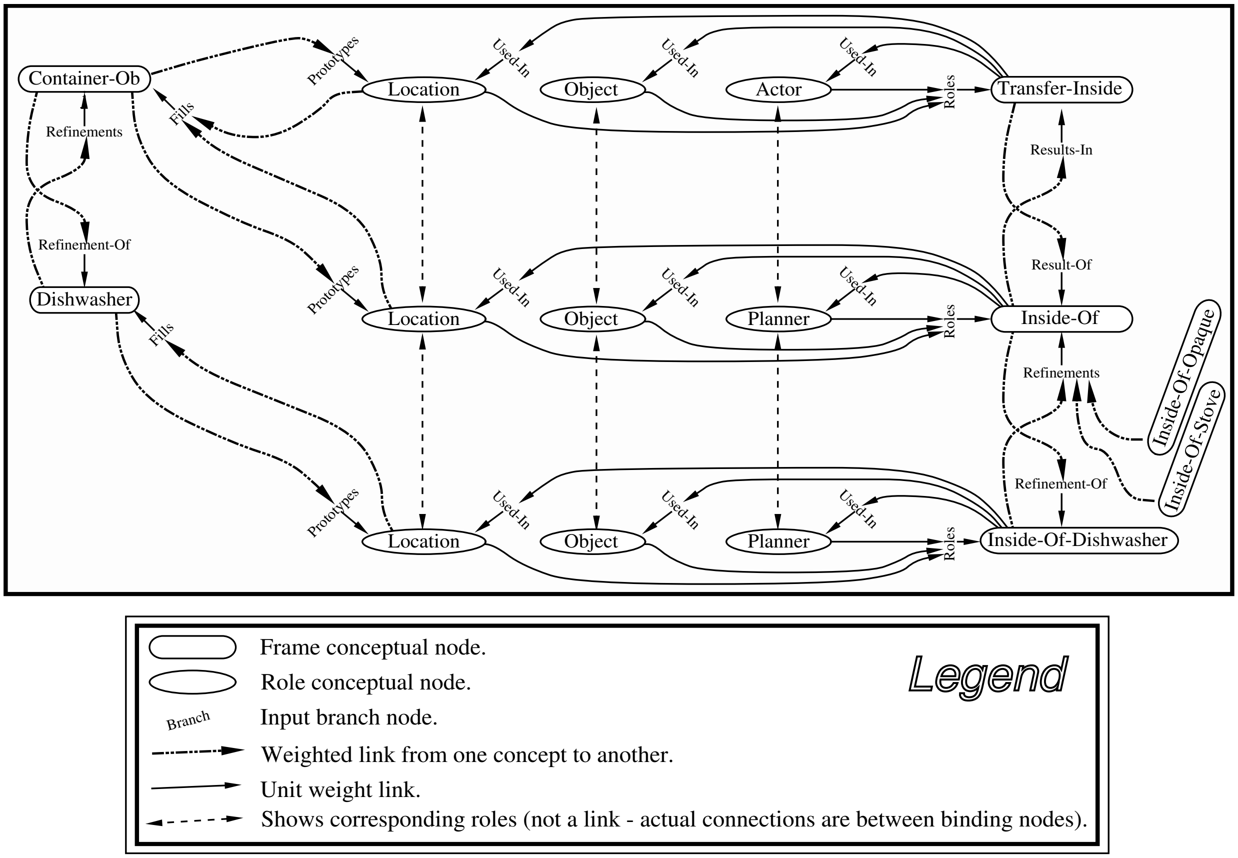


**Figure 8.** Detailed view of the evidential portion of the network for frame Inside-Of and part of frames Transfer-Inside and Inside-Of-Dishwasher. As usual, the node names are used only for initial construction of the network, and do not affect the spread of activation.

branch for frames. With these evidential connections, the Location role of Inside-Of-Dishwasher, for example, will become activated if either its frame (Inside-Of-Dishwasher) or its prototypical filler (Dishwasher) is activated (Figure 8).

Connections providing evidence from the activation of a frame's roles pass through its Roles input branch (as do the Location, Object, and Planner roles of Inside-Of in Figure 8). However, the Roles input branch does not calculate its activation as the maximum of its inputs, because there is no competition between roles. Their activation simply provides evidence for their frame. If none of the roles of a given frame are active, then that frame should receive no evidence from its roles. If all of them are active, then the frame should get maximum role evidence. Ratios in between should provide proportionate evidence. The amount of this evidence should not vary with the number of roles that a frame has: a frame with only a single role should receive just as much activation evidence if its one role is active as a frame with many roles that are all active. Hence the frame's Roles input branch calculates its activation as the *average* of its inputs.

### 4.3. Virtual Structure of the Evidential Network

The connectivity of ROBIN's conceptual layer (e.g. Figure 8) and that of previous structured disambiguation models encodes static relations between frames and their roles' prototypical fillers without making any assumptions about the *actual* role-bindings being processed at any given time. This structure causes each frame node to accumulate an amount of evidential activation that is appropriate *if the actual role-bindings are not known*.

However, in ROBIN, the actual role-bindings usually *are* known (as signatures). Inferences and evidence from actual frame instantiations can be quite different than those from unfilled frame instantiations. For example, if the network knows that somebody is smoking something, then the network can "guess" that the actual object being smoked is either Cigarette or Marijuana by providing evidential activation to its prototypical fillers. However, if John is smoking a Cigarette, then he is not smoking Marijuana. It doesn't matter that Marijuana is something that can be smoked — if Cigarette is the actual filler, then it, and not Marijuana, should receive evidence from that smoking. To take this into account and limit each conceptual node to evidence accumulation from the actual instances in the network, connections between concepts in ROBIN are gated to control the flow of activation based on their signature bindings. Thus, the actual bindings in a given context cause the network to have a *virtual structure* that combines evidence *as if the network was hand-built for those particular instances*, without modifying the hard-wired "default" structure of the network.

The first part of virtual structure in ROBIN channels evidential activation from concepts to the roles that they fill, and vise versa. As described previously, when the filler of a role is unknown, the role receives activation from its prototypical filler or fillers through its Prototypes branch (Figure 8). At the same time, the unbound role provides evidential activation for its prototypical fillers. However, when a role's binding unit is instantiated with signature activation representing a binding, its Prototypes branch is inhibited so that the role does not receive any activation from or provide any activation to its prototypical fillers. Instead, the role receives activation from the actual concept (or concepts) bound to the role, as represented by the signature binding. To complete the feedback loop, the role provides activation to the concept bound to the role (rather than the prototypical filler). This insures, for example, that Marijuana receives no evidence from the smoking frame when somebody is smoking a Cigarette[8].

The most important part of ROBIN's virtual structure has already been mentioned: the network structure that enforces selectional restrictions by stopping propagation of signatures and evidential activation to frames whose logical binding constraints have been violated. Thus, while the fact that an object is Inside-Of something generally provides some evidential activation to Inside-Of-Stove, the fact that a pot is Inside-Of a Dishwasher does not. There is no chance that the pot was inside of the dishwasher in order to cook something. This is shown in the overview of Figure 6, where Inside-Of-Dishwasher and Inside-Of-Opaque received signatures and evidential activation from Inside-Of, but in which Inside-Of-Stove and the other refinements of Inside-Of whose selectional restrictions were violated did not (see [Lange & Dyer, 1989]). Because of this, the network acts as if it has a virtual structure in which Inside-Of has connections to Inside-Of-Dishwasher and Inside-Of-Opaque, but no connections to its other potential re-

[8]The network structure that gates the spread of evidential activation based on the actual signature instantiations is not important for the purposes of this paper. As everything else in ROBIN, it is all part of the network of connectionist units, and is similar to the structure enforcing selection restrictions [Lange & Dyer, 1989].

finements. However, if signatures on Inside-Of's binding units hold a different instantiation (such as a Cake Inside-Of an Oven), then gating from selectional restrictions will result in a different virtual structure.

### 4.4. 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.

ROBIN, on the other hand, has no inhibitory links between competing concepts. It instead uses a group of nodes which act as a global inhibition mechanism. These *global inhibition* nodes serve to inhibit by equal proportions (short-circuit) all concepts in the network when their average activation becomes too high [Lange & Dyer, 1989]. 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" of [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 ROBIN's global inhibition nodes are *short-circuiting inhibitory*, controlling the spread of activation, but leaving *relative* values of evidential activation unchanged.

Because ROBIN'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 (as opposed to driving the "losers" down to 0 using a winner-take-all network), ROBIN is able to easily perform reinterpretation. When new context that favors an alternative interpretation over a previous one enters the network, it boosts the new interpretation's 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, but in which later evidence from the context of **P2** (*the police were coming*) causes a reinterpretation to Marijuana.

## 5. EXPERIMENTS AND DISCUSSION

ROBIN has been implemented in the DESCARTES connectionist simulator, which allows the flexible simulation of structured heterogeneous networks [Lange *et al.*, 1989]. In this section we give a detailed description of the spreading-activation process for disambiguation and reinterpretation of the **Hiding Pot** example and show an example of how the structure enforcing selection restrictions within the network controls crosstalk. In addition, we present a couple of experiments that compare ROBIN to structured spreading-activation and marker-passing models. Finally, we describe several directions in which we would like to extend the model.

### 5.1. A Detailed Example

As a detailed example of how evidence is combined to perform disambiguation and reinterpretation, consider the processing of input for the sentence *John put the pot inside the dishwasher because the police were coming* (**Hiding Pot**). As described in Section 3.2, a phrase is presented to the network by clamping the lexical concept nodes for each of its words to a high activation value (1.0) and by clamping the appropriate binding nodes to the signatures given by the phrase's syntactic bindings. To roughly simulate the sequentiality of reading, the network is allowed to iterate for 10 cycles between the presentation of each word/role-binding. The clamping for the first phrase of **Hiding Pot** is thus the same as that described for Figure 5a, except that the lexical node for *John* is clamped on cycle 1, the lexical node for phrase <S "put" DO "inside" IO> is clamped on cycle 11, and so on.

Figure 9 shows the levels of evidential activation through time for most of the relevant frames during the spreading-activation process for **Hiding Pot.** The first thing that happens is that Transfer-Inside and Inside-Of become activated, in order, as they are inferred from the phrase (Figure 9a). Note that it takes approximately 10 cycles for activation to propagate from one frame to another through the input branch nodes and the nodes that enforce selectional restrictions. During this time, the three ambiguous meanings of *pot* are differentiated only by the strength of their weights from the *pot* node, since they all can be put inside of things. In this particular network, Planting-Pot has the strongest weight from *pot*, so it is winning up until about cycle 60 (Figure 9e).

At about cycle 50, activation and signatures from Inside-Of reach its refinements (Figure 9b). As described earlier, the selectional restrictions in the network stop Inside-Of-Stove and most of Inside-Of's other refinements from receiving either evidential or signature activation, since a pot inside a Dishwasher does not match their roles' binding constraints. Just as important is that the signature for Cooking-Pot is the only one that reaches Inside-Of-Dishwasher's Object (Figure 5b). Because of this, as Inside-Of-Dishwasher increases in activation, Cooking-Pot receives evidence from it (that Marijuana and Planting-Pot do not), causing Cooking-Pot's activation to start rising around cycle 65. Because Inside-Of-Dishwasher in turn receives activation from the virtual binding of Cooking-Pot, a small feedback loop is created between the two that pushes both higher, including the newly instantiated $Dishwasher-Cleaning frame (Figure 9c). By cycle 100, Cooking-Pot is clearly dominant over the other interpretations of *pot*, and would be chosen as the binding throughout the network were the simulation to be halted at this point.

Inside-Of-Dishwasher also becomes more highly-activated than its competitor Inside-Of-Opaque during these cycles, but its advantage is much smaller. The reason for this can be observed in Figure 10, which details the activations of the input branches through which the frames receive their input[9]. Cooking-Pot quickly becomes more active than Marijuana and Planting-Pot because of the unique role that it fills in the highly-specific dishwasher cleaning frames, so that its Fills input branch provides it with much more activation than theirs do in cycles 60 to well over 100 (Figures 10c-e). On the other hand, the added boost that the ascension of Cooking-Pot provides to Inside-Of-Dishwasher is averaged with the

[9]Recall that each conceptual node's activation function includes part of their previous activation, while being short-circuited (normalized) by the global inhibition nodes. Because of this normalization, the activation of each conceptual node is generally less than the sum of its net inputs, as can be seen by comparing nodes' activations in Figures 9b and 9e with their net inputs in Figure 10.

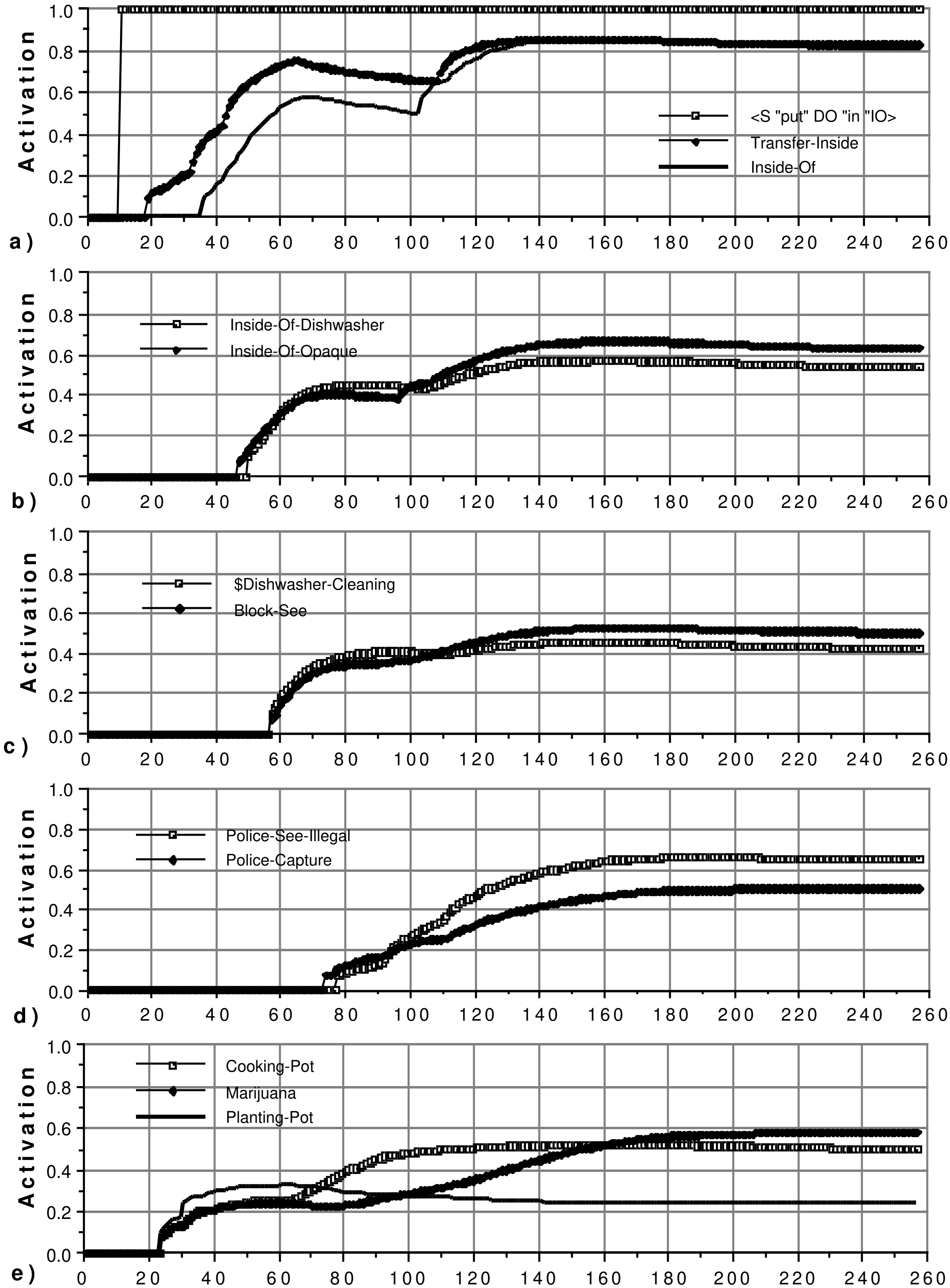


**Figure 9**. Evidential activations in network after presenting input for **Hiding Pot**.