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

INTEGRATING MARKER PASSING AND CONNECTIONISM FOR HANDLING CONCEPTUAL AND STRUCTURAL AMBIGUITIES

Ronald A. Sumida
Michael G. Dyer
Margot Flowers

June 1988
CSD-880044

Integrating Marker Passing and Connectionism for Handling Conceptual and Structural Ambiguities*

Ronald A. Sumida
Michael G. Dyer
Margot Flowers

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

Abstract

This paper** discusses the problem of selecting the correct knowledge structures in parsing natural language texts which are conceptually and structurally ambiguous and require dynamic reinterpretation. An approach to this problem is presented which represents all knowledge structures in a uniform manner and which uses a constrained marker passing mechanism augmented with elements of connectionist models. This approach is shown to have the advantage of completely integrating all parsing processes, while maintaining a simple, domain-independent processing mechanism.

1. Introduction

A major problem in parsing natural language texts is the selection of the correct knowledge structures from the large number of inappropriate ones in memory. This problem is especially difficult in the case of texts which are highly ambiguous and which require the reader to correct an initially mistaken interpretation, since structures which are only potentially relevant must also be found. Consider, for example, the following sentence:

S1. John put the pot on the stove.

This seems to indicate that John is preparing to use a container for cooking on a stove. However, after reading the next sentence:

S2. He picked it up and smoked it.

it appears that John was actually using the stove as a supporter (or lighter) for a marijuana

* This research is supported in part under a contract to the second two authors by the JTF program of the DoD, monitored by JPL and by an ITA Foundation grant to the second author.

** A slightly edited version of this report appears in the Proceedings of the Tenth Annual Conference of the Cognitive Science Society, August, 1988, Montreal.

cigarette (not a cooking pot). In addition note that S1 and S2 are potentially ambiguous at the structural level, e.g. <X picked it up> could mean <X learned new information>, while <X put object on> could mean <X wear object>.

Previous approaches to parsing natural language texts have largely been unsuccessful at handling ambiguous sentences such as those presented above. These approaches can generally be divided into four groups: (1) **Expectation-based conceptual analyzers (CAs)**, such as [Dyer,1983], associate each word with one or more knowledge structures, which have rules attached indicating how they can be connected to other structures. This approach has been successful for parsing large pieces of connected text. However, the processing mechanism is overly complex, since each type of knowledge structure generally requires its own set of rules. Parsing highly ambiguous sentences such as S1 and S2 above is particularly problematic since sophisticated back-up and recovery rules are needed. (2) **PDP/Connectionist systems**, examples of which include [Waltz and Pollack,1985], [Cottrell and Small,1985], [McClelland and Kawamoto,1986], have emerged as an alternative to such rule-based approaches. These systems use only simple rules for spreading and combining activation (and in some cases inhibition). Since they are highly parallel and employ scalar activations, complicated backtracking rules are not needed. Unfortunately, these models currently lack operations which are fundamental in higher level NLP systems, specifically: variables, role bindings, instantiations, and inheritance. (3) **Marker passing systems** [Charniak,1986], [Granger et. al, 1986] and [Norvig, 1987], which find connections between concepts by propagating markers over a local semantic network, are a similar approach which provides these high-level operations. Such systems however, generate too many inappropriate connections and typically employ a filter mechanism with its own set of inference rules to weed them out. The complexity of this mechanism negates the simplicity that is the advantage of the marker passing approach. (4) **Definite Clause Grammars (DCGs)**, such as [McCord, 1982], unlike the above approaches, focus primarily on the syntactic and structural features of natural language texts, such as conjuncts, quantifiers and agreement. These systems view parsing as a two step process which first constructs a syntactic parse tree through unification and then performs semantic processing. The strength of these systems is their ability to analyze complex linguistic constructs. However, they lack the conceptual information necessary to analyze texts at deeper conceptual levels.

This paper presents CAIN (Conceptual Analyzer for multiple re-INterpretations), which parses highly ambiguous texts while avoiding the problems of the above approaches. CAIN overcomes these problems by: (1) representing all knowledge (both conceptual and structural) in a uniform manner in a local semantic network, (2) using constrained marker passing for all parsing processes, and (3) using link weights, activation values, and thresholds from connectionist models for indicating relative strengths of activations between concepts. Representing all knowledge at the symbolic level provides higher level symbolic operations and allows all parsing processes to be integrated. The marker passing mechanism depends only upon knowledge of the different link and marker types used, so the processing mechanism is simple and independent of the content of memory. Also, since only certain types of marker intersections are considered important and since elements of connectionist models are employed, the problem of spurious connections is avoided. CAIN is implemented in T [Slade, 1987], a Scheme-based dialect of Lisp, and can parse sentences S1 and S2 above.

2 Process Model

2.1 Memory Organization

All knowledge, both conceptual and structural (syntactic), is represented using a semantic network, examples of which are shown in figures 1 and 2*.

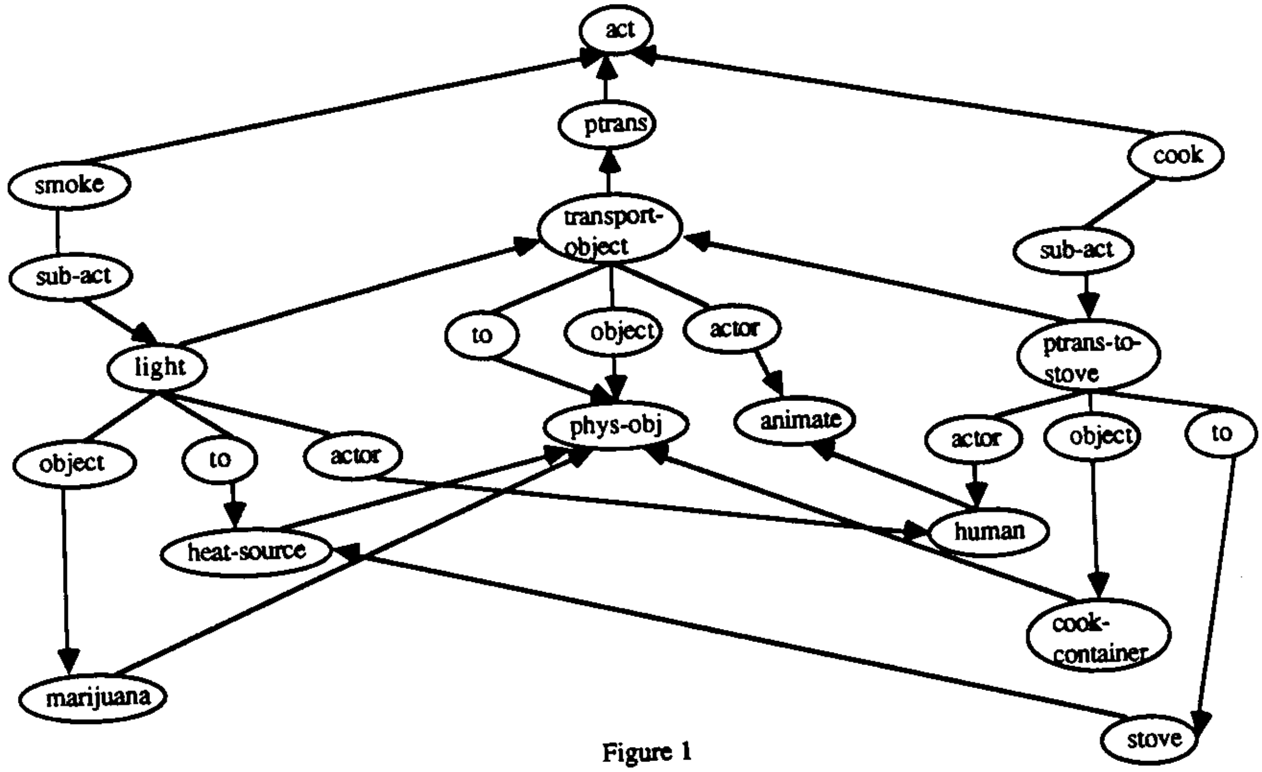

Figure 1

Figure 1 shows the semantic representation, used in parsing sentence S1, for putting a cooking container on a stove and for lighting a marijuana cigarette. An is-a link, which represents an inheritance or category membership relation, is indicated in the figure by an arrow from the child to its parent. For example, to represent that smoking is a type of act, the SMOKE node is connected by an is-a link to the ACT node. A has-a link, which connects a node (called the head or owner) to the nodes which "belong" to it (called the roles), is represented by a straight line in the figure. Both links are bidirectional, so that it is possible for parents to access their children and for roles to access their owners. Slot-filler relationships in traditional frame-based systems are represented by connecting the head node (representing the frame) to its role (representing the slot) by a has-a link, and the role to its filler by an is-a link. For example, to represent that lighting a marijuana cigarette is a sub-act of smoking, the node for SMOKE is connected by a has-a link to its sub-act role, and by an is-a link to the node representing the lighting action. The components of a single act are represented in the same manner. To indicate that the object

* Due to space limitations, the figures in this paper have been simplified and only show the small portion of the network which is activated by parsing S1.

of a transport action is a physical object, TRANSPORT-OBJECT is connected by a has-a link to its object role, which is in turn connected by an is-a link to PHYS-OBJ.

The right side of figure 2* shows the structural information needed for parsing sentence S1.

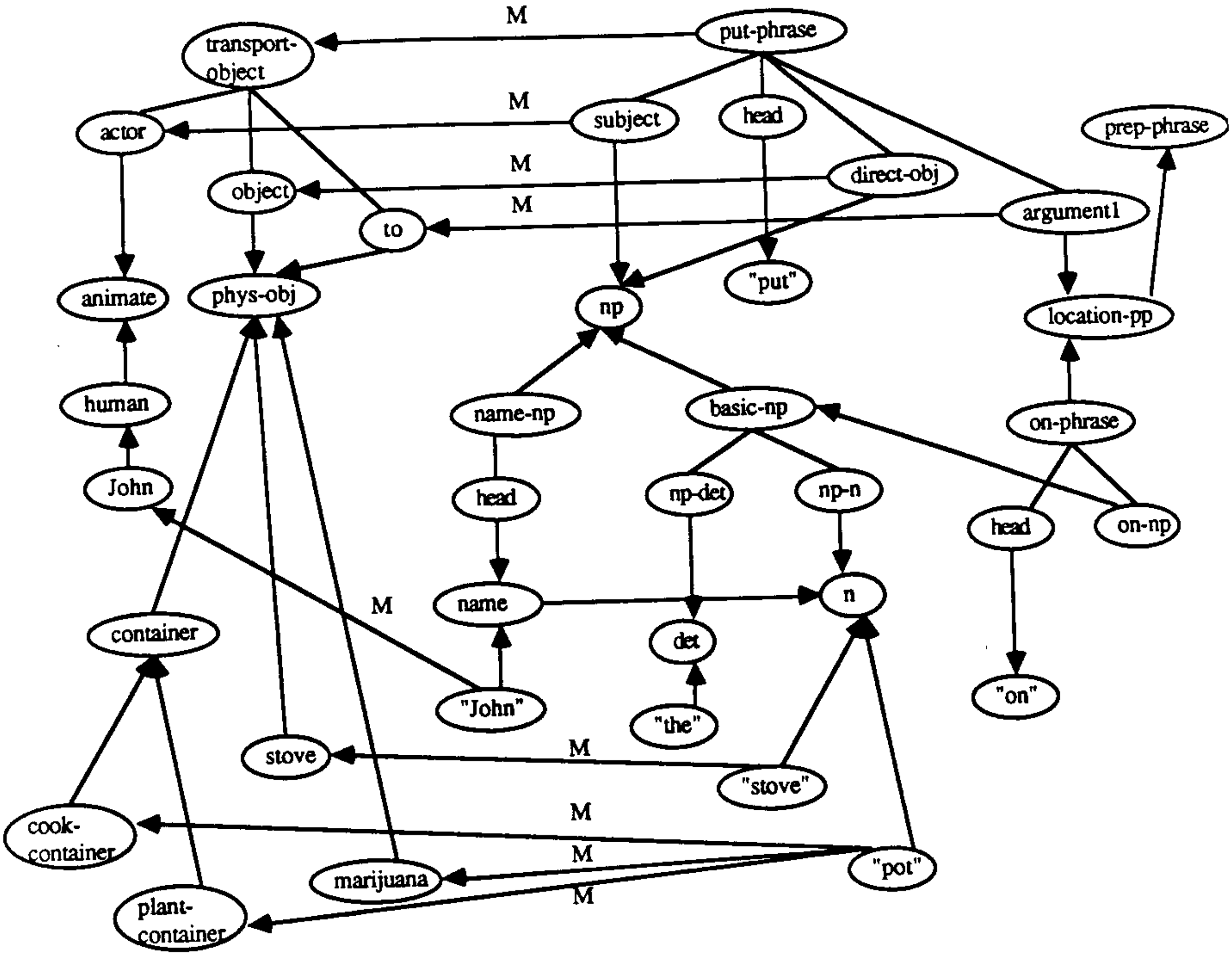

Figure 2

Note that relationships between the syntactic elements in the figure use exactly the same format and link types described above for representing conceptual information. The figure shows the representation for the phrase <person put OBJ1 on OBJ2>, which is indicated in the figure by the put-phrase node. PUT-PHRASE has four roles: the subject role which is a noun-phrase, the head role which is the lexical node "put", the direct object which is also a noun-phrase, and an argument which is a prepositional phrase referring to a location. In sentence S1, the subject is a noun phrase consisting merely of the name "John". The NAME-NP node, whose head role is

* This representation is based primarily upon [Gasser, 1988] and [Jacobs, 1985].

a name, is used to represent this type of NP. The direct-object of S1 is a noun phrase composed of the determiner "the" followed by the noun "pot". This type of noun phrase is represented as a BASIC-NP, with one of the roles (NP-DET) being a determiner and the other (NP-N) being a noun. The left side of figure 2 shows the conceptual information which corresponds to the linguistic representation shown on the right. The link between syntactic and conceptual information is indicated by the arrows labelled M (for meaning). For example, since the meaning of <person put OBJ1 on OBJ2> is the transporting of an object to another object, PUT-PHRASE is connected by an M link to TRANSPORT-OBJECT.

2.2 Parsing Using Constrained Marker Passing

The parsing process can be divided into 4 steps:

1. From the input, mark the lexical items and their associated meanings
2. Find the knowledge structures which connect the marked nodes together
3. Bind the roles of these structures
4. Refine them to be as specific as possible

Consider parsing sentence S1 by the steps listed above. First, each of the lexical nodes (indicated by quotes around the labels) and the nodes connected by M links in figure 2 will be marked as the input is read. In step 2, the put-phrase node is selected from the other competing alternatives in memory through the connections shown in the figure. The third step involves binding the subject, direct-object, and argument roles of PUT-PHRASE, and the actor, object, and to roles of TRANSPORT-OBJECT. Finally, TRANSPORT-OBJECT is refined to PTRANS-TO-STOVE in figure 1, while LIGHT is also considered as a possible refinement candidate.

The following sections describe how the above processes are realized using a constrained marker passing and activation mechanism.

2.3 Marking the Input Concepts

As each word of the input is read, some means is needed for marking the concepts which have been encountered. For example, when the word "pot" is read, the node for "pot" in the network should be marked, as well as the nodes representing its meanings: cook-container, plant-container, and marijuana. An *activation marker (AM)* is used for this purpose. An AM is placed on nodes from the input according to the following rules:

R-1: When a word is read, an AM is placed on its corresponding lexical node.
R-2: When a lexical or phrasal node receives an AM,
an AM is passed across an M link to its associated conceptualizations.

When a concept is encountered, its ancestors are also implicitly encountered. For example, since a cook-container is a type of container and a type of physical object, the occurence of the former implies that of the latter. Thus, marking a concept should also result in the marking of its ancestors. This is accomplished by the rule:

R-3: When a node receives an AM, an AM is passed to its parents.

AMs which are passed to ancestor nodes contain information indicating the descendant that was the source of the marker. In addition, AMs from lexical nodes also maintain information indicating their meaning(s). This information will later be used to perform role bindings.

2.4 Connecting the Input Concepts

How can the correct knowledge structures, connecting the input concepts, be selected? Each node which was activated (received an AM) from the input suggests potentially relevant structures based on the various roles that it plays. This is true for both syntactic and semantic information. For example, since a stove plays the role of an instrument in the cooking schema, activating STOVE suggests that COOK may be applicable. Similiarly, activating the node for determiner indicates that BASIC-NP may be appropriate. *Search Markers (SMs)* are used to indicate knowledge structures which are suggested in this fashion. SMs are propagated according to the following rule:

R-4: SMs are passed from an activated node, down is-a links to all of its descendants that are role nodes, and across has-a links to the owners of the roles.

Applying the above rule will result in the marking of the correct knowledge structures. However, a large number of inappropriate structures will also receive SMs. For example, when the lexical node for "put" is activated in sentence S1, the above rule will mark the nodes for other phrases involving "put", such as <person-put-up-with-person> and <person-put-on-clothing>, in addition to marking the node for the desired put-phrase shown in figure 2. The solution to this problem is to utilize elements of connectionist models, specifically link weights, activation values and thresholds. Each SM is assigned a *strength value* which depends upon the *weights of the links* over which it is propagated. In general, nodes representing more specific concepts will pass stronger SMs than their ancestors. Thus, the SM that COOK-CONTAINER passes to PTRANS-TO-STOVE in figure 1 will be much stronger than the SM that PHYS-OBJ passes to TRANSPORT-OBJECT. When an SM is propagated to a node representing a knowledge structure, its strength value is added to that of the other SMs on the node. If their combined strengths exceed the node's *threshold level*, then there is strong evidence that the structure is applicable, and it therefore attempts to bind its roles. *Using activation values and thresholds allows a large number of structures to be suggested, while only a few are actively pursued.*

2.5 Role Binding

Binding a role of a structure involves determining whether its filler is activated. If so, then the concept which activated the filler is bound to the role. To bind the subject role of PUT-PHRASE in figure 2, for example, the NP node is checked to determine whether it was previously activated. If it was, the descendant which activated it is then bound to the subject role. The check for whether the filler has an AM is made using a *Role marker (RM)*, which is propagated according to the following rule:

> R-5: When a node's threshold is exceeded, RMs are passed across has-a links to each of its roles, and up is-a links to the fillers of those roles.

Note that RMs may be used to indicate roles which should already have been filled or which are expected to become filled. In the latter sense, RMs are very similar to the prediction marker used in DMAP [Riesbeck and Martin, 1986]. Role binding is performed by the rule:

> R-6: When an AM and an RM intersect,
> an AM is placed on the role node.

Since an AM maintains information indicating the descendant that was its source, merely placing it on the role has the effect of binding it.

Our confidence in a structure's relevance to the input increases as its roles are bound. The amount of the increase depends upon how important the role is to the structure. Role importance is reflected in the strength of the connection between the structure and its roles and therefore in the strength of the RM which is passed by rule R-5. The rule for activating a structure is:

> R-7: When an RM and an AM intersect, activate the source of the RM
> by placing on it a new AM if one (representing this instance) is not already present,
> or by increasing the activation level of the AM which is already there.

For example, as sentence S1 is read and each component of the put-phrase in figure 2 is recognized, it becomes apparent that it correctly represents the input. Thus, binding the subject role should activate the put-phrase node, binding the head role increases its activation level, and similarly for the remaining roles.

2.6 Concept Refinement

The *most specific* structures possible must be found in order for the input to be completely parsed. A node which is activated by rules R-2 or R-7 may need to be refined to a more specific one using contextual information supplied from the input. Refinement involves searching for a descendant whose equivalent roles have more specific, activated fillers. For example, when the put-phrase is recognized and TRANSPORT-OBJECT is activated by R-2, it can be refined to PTRANS-TO-STOVE as shown in figure 1. The search process is performed using a *descendant marker (DM)* which is spread by the rule:

> DM-1: When a node is sufficiently activated by rules R-2 or R-7, a DM is passed down is-a links to each of its descendants

How is a descendant with more specific fillers found? Recall that each concept which was activated from the input supplies contextual information in the form of SMs, whose strengths are combined when they intersect. A descendant whose fillers are more specific will have SMs with a stronger combined strength. The rule which implements concept refinement is:

DM-2: When a DM is placed on a node whose combined SM level is greater than that of the source of the DM, then bind its roles using the procedure described in section 2.5

For example, PTRANS-TO-STOVE in figure 1 will receive SMs from HUMAN, COOK-CONTAINER, and STOVE while TRANSPORT-OBJECT will receive SMs from ANIMATE, PHYS-OBJ, and PHYS-OBJ. Since the connection from the roles of PTRANS-TO-STOVE will be much stronger than for TRANSPORT-OBJECT, the former will have a much greater SM strength. Thus, TRANSPORT-OBJECT should be refined to PTRANS-TO-STOVE.

2.7 Marker Removal

As with connectionist systems, markers are removed using a decay process. DMs decay very quickly since they do not have to wait for other nodes to become activated and therefore do not need to remain between sentences. This is not true for the other types of markers, so they decay much more slowly.

3. A Detailed Example

This section provides a detailed look at the parsing of sentence S1.

First, the word "John" is read (figure 2), so an AM is placed on the node for "John" by rule R-1 (When a word is read, an AM is placed on its corresponding lexical node). An AM is also placed on NAME by rule R-3 (When a node receives an AM, an AM is passed to its parents). The node for the concept John is activated by R-2 (When a lexical or phrasal node receives an AM, an AM is passed across an M link to its associated conceptualizations), and HUMAN and ANIMATE are activated by R-3. SMs are propagated from each of these marked nodes by R-4 (SMs are passed from an activated node, down is-a links to all of its descendants that are role nodes, and across has-a links to the owners of the roles). Thus, in the figure, NAME passes an SM to NAME-NP through its head role. Since NAME-NP has only one role, its threshold is exceeded and it passes an RM to NAME by R-5 (When a node's threshold is exceeded, RMs are passed across has-a links to each of its roles, and up is-a links to the fillers of those roles). This intersects the AM that was previously placed on NAME, so "John" is bound to the head role by R-6 (When an AM and an RM intersect, an AM is placed on the role node). In addition, an AM is placed on NAME-NP by R-7 (When an RM and an AM intersect, activate the source of the RM by placing on it a new AM if one is not already present, or by increasing the activation level of the AM which is already there) and on NP by R-3. Figure 3a shows how the markers are passed for this portion of memory. The effect of the other SMs that are passed is very small, since the activated nodes are so general.

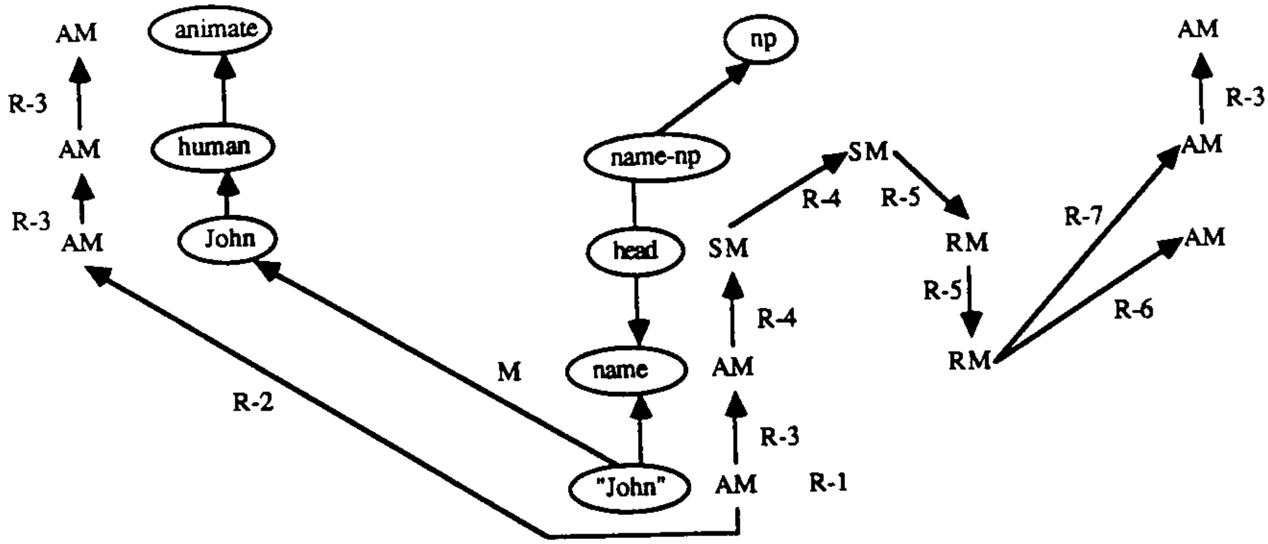

Figure 3a

Next, "put" is read, resulting in the placement of an AM on the node for "put". An SM is passed from "put" to the put-phrase node through its head role. All of the other phrases (not shown in the figure) involving "put" will also receive SMs of varying degrees of strength. Since the put phrase shown in figure 2 is one of the most general and common involving "put", the strength of the SMs it has received is sufficient to exceed its threshold. Rule R-5 thus applies, so RMs are propagated to the roles SUBJECT, HEAD, DIRECT-OBJ, and ARGUMENT1, and to their respective fillers. The RM which is placed on NP from the subject role intersects with the AM from "John", so the AM is placed on SUBJECT (declarative sequencing information, associated with the put-phrase node, causes rule R-6 to apply to the RM from the subject rather than DIRECT-OBJ). By rule R-7, a new AM of relatively weak strength is placed on the put-phrase node. An RM-AM intersection also occurs on the "put" node, so the AM on the put-phrase is strengthed by R-7. The resulting state appears in figure 3b.

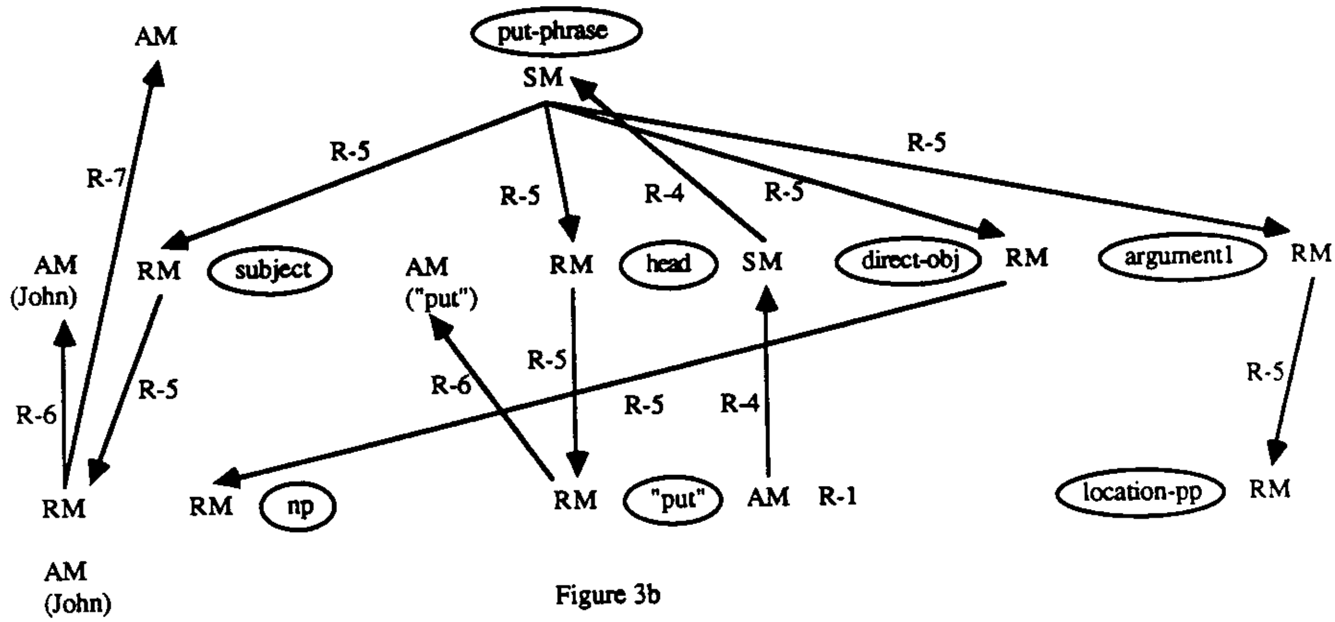

Figure 3b

Reading "the" activates the determiner node, and suggests BASIC-NP. This is sufficient for BASIC-NP to begin binding its roles, so NP-DET receives the AM from "the", and BASIC-NP is weakly activated. The word "pot" is read next, and AMs are placed on "pot", N, its three meanings, and their ancestors. The np-n role is bound to "pot", and BASIC-NP is now strongly activated. By R-3, this activates NP. Rules R-6 and R-7 now apply, binding this instance of the NP to the direct-object role and further strengthening the AM on the put-phrase. Rule R-2 also applies, causing TRANSPORT-OBJECT to receive an AM from PUT-PHRASE, ACTOR from SUBJECT, and OBJECT from DIRECT-OBJECT. As we noted in section 2.3, each AM which is placed on a phrasal node maintains information about its meaning. Thus, the AM which ACTOR receives will have John as its meaning, and the AM on the object node maintains the three meanings of "pot". Thus, syntactic information is used to help bind roles in semantic structures. The resulting memory state appears in figure 3c. Note that in parallel with the above, markers are also spread to conceptual nodes. For example, in Figure 1, COOK-CONTAINER passes an SM to PTRANS-TO-STOVE, which is a sub-act of COOK, and MARIJUANA passes an SM to LIGHT, a sub-act of SMOKE.

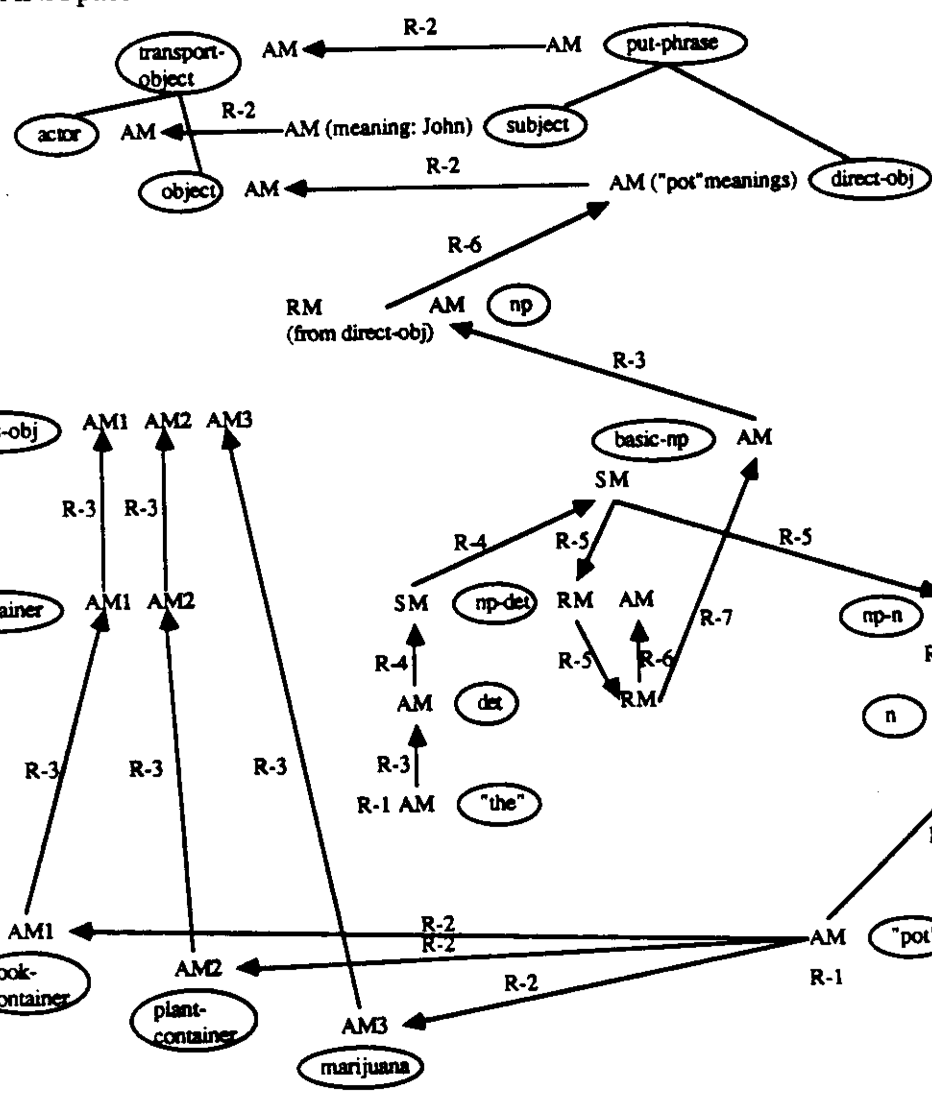

Figure 3c

Parsing "on" suggests the on-phrase. Finally, reading "the stove" activates NP, ON-PHRASE, and LOCATION-PP, which binds the argument1 role of PUT-PHRASE and the TO role of TRANSPORT- OBJECT. In addition, STOVE passes an SM to PTRANS-TO-STOVE and an AM to HEAT-SOURCE, which causes an SM to be passed to LIGHT. Figure 3d shows the AMs which are placed on the structures in figure 2, as a result of reading S1.

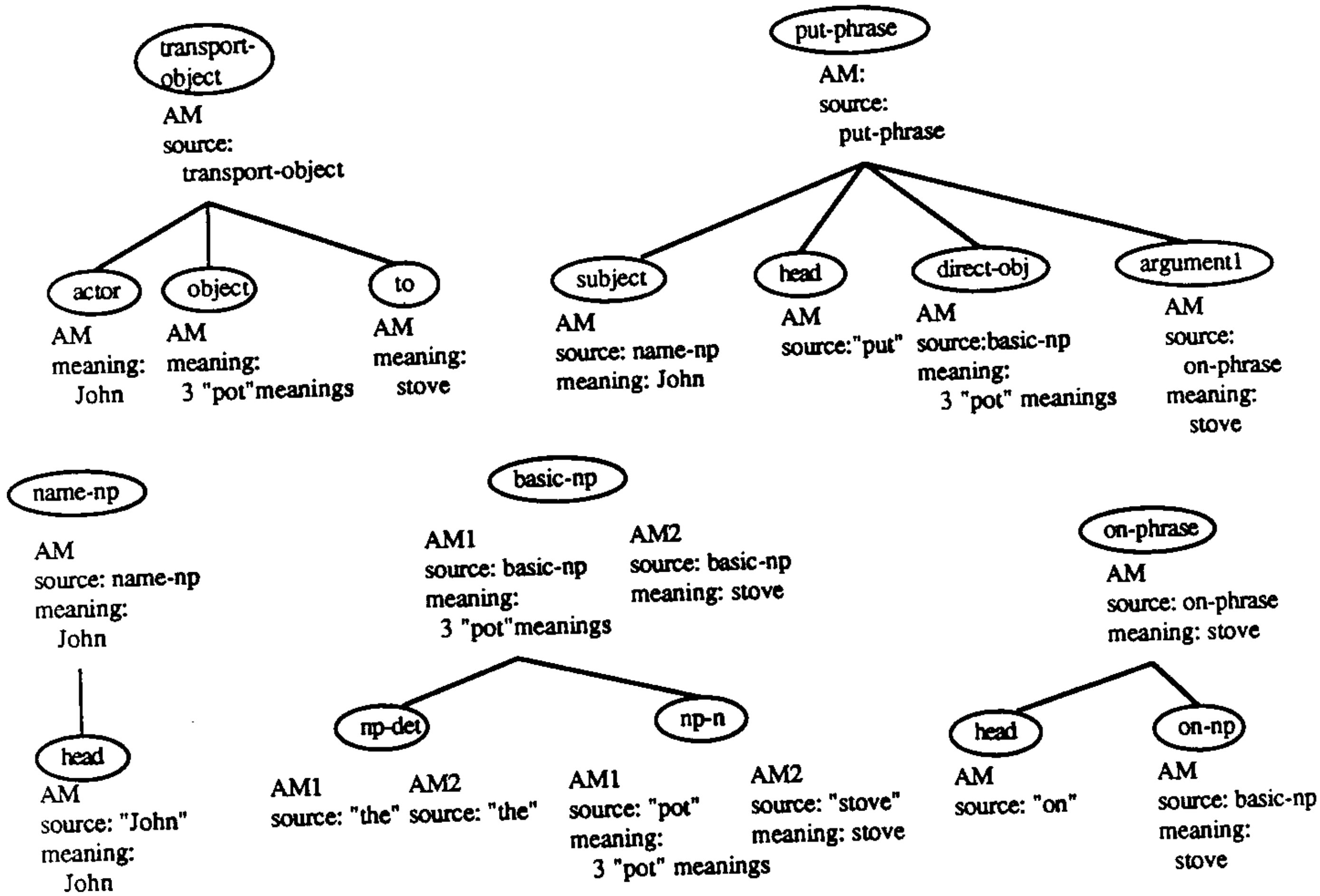

Figure 3d

Concept refinement now occurs. Note that TRANSPORT-OBJECT can be refined to either PTRANS-TO-STOVE or LIGHT, so both will be activated. However, since the concept stove suggests cooking much more strongly than HEAT-SOURCE suggests lighting a marijuana cigarette, PTRANS-TO-STOVE will be much more strongly activated. Therefore, it represents the result of the parse.

When sentence S2 is read, however, it is recognized as another sub-act of SMOKE. SMOKE will therefore be more strongly activated than COOK, since it receives SMs from two of its sub-act roles, while COOK is unrelated to S2 (figure 3e).

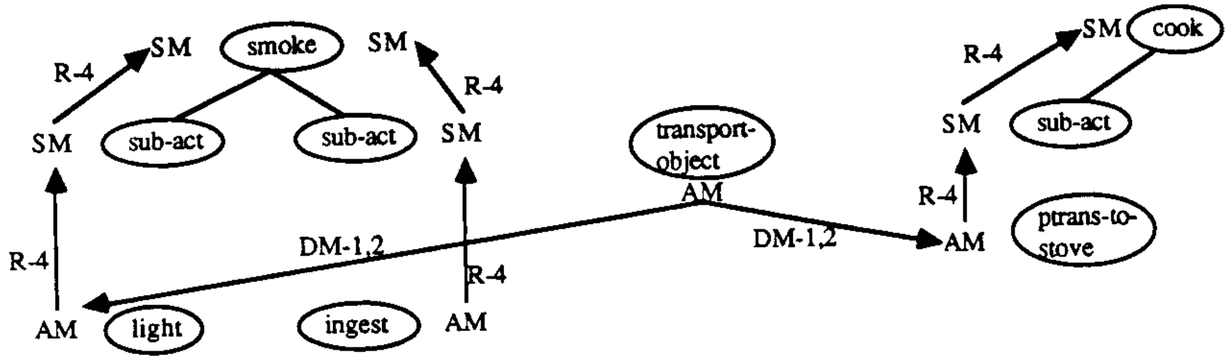

Figure 3e

This causes S1 to be reinterpreted as lighting a marijuana cigarette. It is important to note that as a sentence is being read, the roles are bound with markers for both the correct interpretation and the potentially relevant ones. For example, both PTRANS-TO-STOVE and LIGHT have their roles bound with markers as S1 is read. Performing bindings for multiple interpretations allows CAIN to correctly re-interpret S1 without undoing any previous bindings. Thus, if a third sentence, such as "He did this by putting it, and the ham in it, in the smoker oven.", indicated that S1 really did refer to COOK, CAIN would be able to easily change its interpretation.

4. Related Work

The work presented here was inspired by direct memory access parsing, particularly DMAP [Riesbeck and Martin, 1986]. DMAP attempts to find the most specific knowledge structures that connect the input concepts, using a marker passing algorithm based on recognizing concept sequences. DMAP uses two types of markers: (1) *activation markers (A-markers)*, which are placed on lexical nodes from the input, on their ancestors, and on structures whose entire concept sequence has been encountered and (2) *prediction markers (P-markers)*, which indicate the next element of the concept sequence that is expected to occur. Every knowledge structure starts with a P-marker on the first element of its concept sequence. Refinement occurs when an A-marker and a P-marker intersect, and a special concept refinement link is used for sending the P-marker to the more specialized source of the A-marker.

Despite the similarity between DMAP's markers and ours, there are major operational differences. The biggest difference is that DMAP is only able to recognize structures whose roles are encountered in the correct sequence, beginning with the first item. While this works well for syntactic structures which are typically encountered in their entirety and in the correct order, it is not well suited to recognizing higher level conceptual structures such as MOPs [Schank, 1982]. For example, DMAP would not be able to recognize that the COOK context is appropriate after parsing sentence S1, since the intial act, PTRANS-FOOD-TO-CONTAINER, was not encountered. Our work also extends direct memory access parsing (1) to handle ambiguities, reinterpretations, and role bindings, (2) to include more information about syntax, and (3) to represent relative strengths of activations between concepts.

We believe that learning (i.e. adding new nodes and links to the network) will be facilitated by the simplicity of our memory representation. This contrasts to approaches which simplify the processing mechanism by adding extra link types to the network (such as the special concept refinement link used in DMAP). Our approach is to use only those link types which are necessary for the representation itself and add extra markers where necessary. Since markers are dynamically created during processing and decay with time, adding new ones has no effect on the complexity of the learning mechanism. Similarly, link weights in our model *only* represent relative strengths of connections between concepts and (unlike connectionist systems) are *not* used to control processing.

This work also bears some similarity to SCISOR [Rau, 1987], a system for conceptual information retrieval. SCISOR receives pre-parsed questions about short newspaper stories, searches memory (using a marker passing mechanism) for episodes which may provide the answer, evaluates these episodes, and outputs the answer using a separate generator. The process presented in section 2.4 (for finding the correct structures in memory connected to input concepts) is similar to the priming rules used in SCISOR. However, SCISOR only addresses the problem of finding episodes in memory, and uses a separate module for parsing. In our work, *parsing and memory search are completely integrated.* Thus, the memory search process described here is more general since SMs can be used to retrieve different types of knowledge structures (such as syntactic information) in addition to retrieving episodes.

5. Future Work

Two important learning issues remain to be addressed. The first concerns how new information is added to memory. Each instance of a concept in our model is represented by a unique activation marker. At some point in the parsing process, the long-term memory representation must be modified to include the new instances. The mechanism which performs this modification must decide which of the activated concepts to add to long-term memory and when and where to add them. The second issue concerns how relationships between existing concepts are changed. We are currently investigating how automatic learning methods (similar to those used over distributed representations) can be used at the localist level to modify connection weights through training. These two issues remain unresolved in all marker-passing and localist connectionist models.

6. Conclusions

In this paper, we have presented an approach to parsing natural language texts which integrates a constrained marker passing mechanism with properties of connectionist systems: link weights, activation values and thresholds. This approach is particularly attractive for three reasons. First, it is capable of parsing texts which have proved to be difficult for previous parsing systems, specifically those which are highly ambiguous and require the reader to correct an initially mistaken interpretation. Second, it uses a simple processing mechanism whose rules are independent of the actual content of memory. Thus, new knowledge structures can be added without changing the processing mechanism. Finally, it completely integrates all parsing processes, such as memory search, disambiguation and inferencing.

References

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

Cottrell, G. and Small S. A connectionist scheme for modeling word-sense disambiguation. *Cognition and Brain Theory, 1*, 89-120, 1985.

Dyer, M.G. *In Depth Understanding*. Cambridge, MA: The MIT Press, 1983.

Gasser, M., *A Connectionist Model of Sentence Generation in a First and Second Language*. UCLA PhD. Thesis, forthcoming.

Granger, R.H., Eiselt, K.P. and Holbrook, J.K., Parsing with Parallelism: A Spreading Activation Model of Inference Processing During Text Understanding. In J. Kolodner and C. Riesbeck (Eds.), *Experience, Memory, and Reasoning*. Hillsdale, NJ: Lawrence Erlbaum, 1986.

Jacobs, P. *A Knowledge-Based Approach to Language Production*. UC Berkeley PhD. Thesis, 1985. Computer Science Division Report UCB/CSD86/254.

McCord, M.C. Using slots and modifiers in logic grammars for natural language, *Artificial Intelligence*, 18 (1982), 327-367.

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

Norvig, P., Inference in Text Understanding, *Proceedings of the National Conference on Artificial Intelligence (AAAI-87)*, Seattle, WA, 1987.

Rau, L.F., Spontaneous Retrieval in a Conceptual Information System. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA, 1987.

Riesbeck, C.K. and Martin, C.E. Direct Memory Access Parsing. In J. Kolodner and C. Riesbeck (Eds.), *Experience, Memory, and Reasoning*. Hillsdale, NJ: Lawrence Erlbaum, 1986.

Schank, R.C. *Dynamic Memory: A theory of Reminding and Learning in Computers and People*. NY: Cambridge University Press, 1982.

Slade, S. *The T Programming Language*, Englewood Cliffs, NJ: Prentice-Hall, 1987.

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