Appears in Computational Intelligence, Volume 13, Number 1, 1997

DETERMINING PREPOSITIONAL ATTACHMENT, PREPOSITIONAL MEANING, VERB MEANING AND THEMATIC ROLES

FERNANDO GOMEZ, CARLOS SEGAMI, AND RICHARD HULL
Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA

An algorithm for semantic interpretation that integrates the determination of the meaning of verbs, the attachment and meaning of prepositions, and the determination of thematic roles is presented. The parser does not resolve structural ambiguity, which is solely the task of the semantic interpreter. Lexical semantic information about nouns and verbs is applied to the resolution of verb polysemy and modifier attachment. Semantic interpretation is centered around the representation of the meaning of the verb, called verbal concept. Verbal concepts are organized into a classification hierarchy. As long as the meaning of the verb remains unknown, parsing proceeds on a syntactic basis. Once the meaning of the verb is recognized, the semantic component makes sense of the syntactic relations built so far by the parser and of those still to be parsed. The algorithm has been implemented and tested on real-world texts.

Key words: natural language processing, semantic interpretation, prepositional attachment, thematic roles, verb polysemy.

1. INTRODUCTION

We present a model for semantic interpretation that integrates the determination of the meaning of verbs, the attachment and meaning of prepositions, and the determination of the thematic roles. The model occupies a middle ground between those approaches that rely heavily on the parser for building structures and attaching PPs, subordinate clauses, etc. (Grishman et al., 1991; Hobbs 1991; Tomita 1985) and semantic-centered approaches (Riesbeck and Martin, 1986; Cardie and Lehnert, 1991; Slator and Wilks, 1991; Wilks et al., 1989). The resolution of structural ambiguity (attachment of PPs, relative clauses, subordinate clauses, etc.) is entirely the task of the semantic interpreter. Thus, for a highly ambiguous sentence like *I saw the man on the hill with the telescope*, the parser will produce only one parse! Regarding the attachment of PPs, the proposed method falls squarely within the semantic approaches (Wilks, 1975) and differentiates itself not only from syntactic approaches, but also from syntactic-semantic ones (Schubert, 1986). An essential concern of the model is to determine the meaning of PPs, not only their attachment, since attaching a PP without determining its meaning is of little or no use to the understander. In that regard, the different syntactic parsing preferences (Right Association (Kimbal, 1973), Early Closure (Kimbal, 1973), Late Closure (Frazier and Fodor, 1978), Minimal Attachment (Frazier and Fodor, 1978) and others) are of little value since they say nothing about the meaning of the PP. Moreover, recent studies of corpora (Whittemore and Ferrara, 1990) have shown that no single preference attachment heuristic that they considered correctly determined PP attachment in more than 55% of 725 sentences containing PP attachment ambiguities. It seems that what works better is right association combined with specific lexical information (Ford et al., 1981; Gibson and Pearlmutter, 1993; Hindle and Rooth, 1991; Hobbs and Bear, 1990).

A claim of the model is that there is a strong relation between determining the meaning of the PP and its attachment, as is clearly illustrated by the sentences *Peter ate the cake with a fork*, and *Peter ate the cake with a cherry.* This is also true for attaching PPs in a string of NPs. For instance, in the phrases "the children in the bus with good brakes" and "the children in the bus with brown hair," a meaningful relation may be found between "brakes" and "bus" resulting in attaching "with good brakes" to "bus," but no relation can be found between "hair" and "bus," while there is a relation between "hair"and "children," resulting in attaching "with brown hair" to "children," violating right association.

Because our model hinges on the representation of the meaning of the verb, mechanisms intended to determine its meaning are an essential component of the model. Most semantic approaches to prepositional attachment assume that the meaning of the verb is already known. However, this is not a realistic assumption because in many cases the meaning of the verb remains unknown until the very end of the sentence. Consider *Peter took an aspirin* vs. *Peter took an aspirin to Mary.* Our parser (Gomez, 1989) is a lexically-driven algorithm that incorporates mechanisms to handle general syntactic phenomena such as relativization, long-distance dependency, passive sentences, etc. However, the enormous syntactic diversity of most words in English is handled by storing their syntactic subcategorizations represented as transition diagrams in the lexical entries of the words. The parsing of a sentence is driven by the syntactic subcategorizations that are retrieved by the parser as they are encountered in the sentence. As explained, modifiers are left unattached in the structure produced by the parser for the sentence. Relative clause referents are not resolved either. The parser builds all the possible referents and lets the semantic component choose the correct one. If the sentence is ambiguous, other than for attachment, e.g. *I saw her duck*, the parser produces all parses for that sentence, and the semantic component selects the correct one. However, as the reader will see below, our *interpretation algorithm is independent of the parsing method adopted* and demands little from the parser, which does not resolve structural ambiguity.

This paper is organized as follows: the second section describes briefly the rules that recognize the meaning of the verb; the third presents the representation of the verbal concepts and the processing by the interpreter after the meaning of the verb has been recognized; the fourth describes the algorithm for attaching prepositions and determining their meaning, and explains its main steps; the fifth discusses some of the main issues posed by the method; the sixth explains the testing of the algorithm in real-world texts; the seventh presents the related research; and the eighth section gives our conclusions.

2. RULES TO DETERMINE THE MEANING OF THE VERB

If rules for recognizing the *meaning of the verb* (henceforth called *verbal concept*) are defined, the parser interacts with a semantic interpretation algorithm, which will be explained below. The parser continues parsing in the same manner as prior to the recognition of the meaning of the verb, but the syntactic relations are analyzed by the interpretation algorithm, which determines the thematic role and/or the attachment of prepositional phrases. ¿From then on, the structure produced by the parser can be discarded; although, in our present implementation, we have kept both outputs for testing and verification purposes (see appendix). We will use the following syntactic relations throughout the paper: PP, subject, object1, object2, predicate. Object2 is built for the second post-verbal NP of verbs that may take two NP complements. These include a diverse list of verbs such as "give," "call," "vote," "consider," "throw," etc. The parser also produces an analysis of verb adjuncts like *time* NPs, and *distance* NPs, e.g., *Peter read every hour*, *Geese fly long distances.*

The rules that determine the meaning of the verbs, henceforth called VM rules, are classified as *subj-rules, verb-rules, obj1-rules, obj2-rules, pred-rules*, and *prep-rules.* These rules are activated when the corresponding syntactic relations are parsed. A *verb-rule* is a rule that is fired when the verb is parsed; an *obj1-rule* is a rule that is fired when the first post-verbal NP is parsed, an *obj2-rule* is a rule that is activated when the second post-verbal NP is parsed, etc. If none of the rules fires, the parser inserts the syntactic relation in the structure being built. This syntactic action performed by the parser may be viewed as a catch-all rule that builds a syntactic relation if no VM rule fires. *Prep-rules* are, in turn, identified by prefixing them with the preposition. Thus, *to-rules* for "take" refer to all those VM rules that will be activated when a prepositional phrase introduced by "to" is parsed in a sentence or clause containing the verb "take." These rules are standard *if-then* rules. If the antecedent is true, then the consequent is executed. An example of an *obj1-rule* for the inflected form of "give" is:

R1. if concept (np) is-a LTM(infection) then meaning-of-verb = transmit-infection; thematic-role-of-obj = theme.

This rule says that, if the concept denoted by *object1* is a subconcept of *infection*, then the meaning of the verb is *transmit-infection*. This rule is activated when the parser has parsed the post-verbal NP in the sentence, say, *Mary gave the measles to Jennifer*. Had this rule not existed, the parser would have built an *obj1* and continued parsing. Besides establishing the meaning of the verb, the rule recognizes the first complement of the verb as the *theme* role. The other syntactic relations already parsed and those still to be parsed are mapped into the thematic roles by an algorithm, explained in a later section, which compares the representation of the meaning of the verb and the syntactic relations.

All those concepts that are Long-Term Memory categories are indicated by writing LTM(concept). Thus, the algorithm that fires the rules searches LTM concepts to verify the antecedents. The main reasons for this are to achieve maximum generality in the application of the rules, and, also, to automatically extend the application domain of the rules by acquiring knowledge by being told. Thus, if our present system is told that *Mumps are infections*, it will be able to use the rule above to make sense of *Mary gave the mumps to Peter*. A minor drawback is that, to set up the rules, a user needs to be familiar with the initial ontology of our system.

For some verbs, it is practically impossible to set up VM rules in such a way that the verbal concept will be recognized prior to reaching the end of the sentence. Consider *Peter took an aspirin*. One could write an *obj1-rule* for "take" saying:

R2. If concept(np) is-a LTM(medicament) then meaning-of-verb = ingest.

The problem with that rule is shown by the sentence *Peter took an aspirin to Mary*. It would not be very hard to accommodate in our algorithm mechanisms that recuperate from wrong inferences produced by VM rules. This would be accomplished by writing rules that override other rules. However, our present and working solution has been to define a class of rules, called *end-of-clause* rules. *End-of-clause* rules are only fired if the parser reaches the end of a clause and the meaning of the verb remains unknown. Accordingly, instead of defining rule R2 above as an *obj1-rule*, we define it as an *obj1-rule* marked as an *end-of-clause* rule. Thus, if the parser reaches the end of the sentence and the verbal concept remains unrecognized, the algorithm will retrieve the *object1* built by the parser and will apply rule R2 above and any other *obj1-rule* marked as an *end-of-clause* rule. On the other hand, the sentence *Peter took an aspirin to Mary* is handled by a *to-rule* for "take" saying:

R3. If concept(object-of-prep) is-a LTM(animate) and concept(subj) is-a LTM(animate) then meaning-of-verb = transport.

The verbal concept *transport* is a subconcept of the primitive *ptrans* (the transfer of the physical location of an object) (Schank, 1975). *Transport* is identical to *ptrans*, except that it has a *co-theme* and a *recipient*, or *goal*, which is optional. When the prepositional phrase "to Mary" is parsed, rule R3 is activated and fires, determining the verbal concept. As a consequence, rule R2 will not be tried at the end of the clause. However, in the sentence *Peter took an aspirin*, rule R2 is fired at the end of the sentence, determining the meaning of the verb. Thus, our solution to the problem of conflictive semantic cues is to procrastinate (Rich et al., 1987) until there is absolute certainty that a rule determines the meaning of the verb. As a result, a considerable number of rules for highly ambiguous verbs are going to be marked as *end-of-clause* rules. For a solution consisting of ranking a diverse class of preferences (syntactic, semantic, collocation, clusters and frequency of information) in determining the meaning of words in large corporas, see (McRoy, 1992). As a final example, consider the following *obj2-rule*, which determines the meaning of "took" in *Peter took Mary an aspirin.*

R4. if concept(np) is-a LTM(physical-thing) and concept(obj1) is-a LTM(animate) then meaning-of-verb = transport.

VM rules solve lexical ambiguity within the process of semantic interpretation, showing that the syntactic realization of the words play a determinant role in resolving lexical ambiguity. In (Gomez, 1995) we explain how VM rules are used to select between the different senses of nouns.

3. PROCESSING AFTER THE MEANING OF THE VERB HAS BEEN DETERMINED

When the verbal concept has been recognized, the syntactic relations already identified by the parser and any subsequent ones are interpreted by comparing them against the semantic representation of the verbal concept. This comparison consists of searching for subsumption relations between the entries in the representation of the verbal concepts and the head noun of the syntactic relations. Verbal concepts corresponding to the senses of "drive," "walk," etc. are not immediately reduced to *primitives*, say *ptrans*, but they are represented in a classification hierarchy.

Figures 2 and 3 depict a very small (the actual hierarchy in our system takes over eight pages) portion of the hierarchy for the family of *ptrans* verbal concepts, which we have called *ptrans-1*, in which the *actor* of the verbal concept is also its *theme*. This hierarchy does not include those *ptrans* verbal concepts, such as some senses of "put," "pour," "send," etc., in which the *actor* is not the *theme* of the verbal concept. This family of *ptrans* verbal concepts form a taxonomy of their own. Figure 1 depicts portions of the general concept *action*, which is the root node of the action verbal concept hierarchy.

The entries in the node *action* have been greatly simplified. The *ptrans-1* verbal concepts form a hierarchy whose root node is *action* and whose parent node is *ptrans-1* and whose sibling nodes are other *ptrans-1* verbal concepts. The differences between, say *drive*, and its parent node, *ptrans-1*, are the slots *actor* and *instrument*. The *actor* of *drive* must be subsumed by the concept *human*, and its *instrument* must be subsumed by the concept *motor-vehicle*. All the other slots for *drive* will be inherited from its superconcepts, *ptrans-1* and *action*. Note that the entry for *drive* does not recognize that in the sentence *Pat drove to school with Kelly* it is unclear whether "Pat" or "Kelly" is the *actor* of the verbal concept. [1] This type of sentence needs to be handled by a VM rule that treats "drive" as ambiguous. One needs a *with-rule* for "drive" that fires if the object of the preposition is subsumed by the concept *human*, and that checks context in order to resolve the ambiguity. If context does not help, then the logical form needs to represent this ambiguity (this logical form has been called a *quasi-logical* form), until subsequent sentences resolve it. Consider the short discourse, *Peter drove to school with Kelly. She drove her new car.*, which is more or less - we prefer to be vague rather than to be accused of not reading Quine - synonymous to *Kelly drove Peter to school in her new car.* Of course, there are many other situations in which VM rules need to delay commitment beyond points where the meaning of the verb can be

[1] We are indebted to an anonymous referee for this example.

```
(action
 (subj (ltm-ctgy (animate (actor))
                 (organization (actor))))
 (obj1 (ltm-ctgy (physical-thing (theme))))
 (prep
  (at (ltm-ctgy
       (location (at-loc (strong)))           ; peter met mary at the beach
       (time (init-time (strong)))))          ; peter smoked at 4 o'clock
  (for (ltm-ctgy
       (human (beificiary (weak)))            ; peter ate the cake for mary
       (time                                  ; peter talked for 2 hours
          (duration (strong)))))
  (in (ltm-ctgy
       (countable-time-unit                   ; peter made the trip in 10 days
          (duration (strong)))
       (time (at-time (strong)))              ; peter took the trip in may
       (manner (manner (strong)))             ; peter shouted in fear
       (physical-thing (at-loc (strong)))))   ; peter studied the book in the
                                              ;   library
  (on (ltm-ctgy
       (time (at-time (strong)))              ; peter listened on monday
       (physical-thing (at-loc (weak)))))     ; peter sang on the doorstep
  (by (ltm-ctgy
       (time (end-time (strong)))             ; peter cooked the spaghetti
                                              ;   by 8 pm.
       (physical-thing (at-loc (weak)))))     ; peter pushed the button by
    .                                         ;   the door.
    .
    .
 )
 (time-np ... (see Gomez, 1995 for a discussion of this))
)
```

Figure 1. Action Node of the Hierarchy

safely determined, because the thematic roles cannot be recognized with certainty. Consider *Peter rented a car to Mary* vs. *Peter rented a car from Mary*. As a consequence, only context can disambiguate *Peter rented a car* with complete certainty. See (Palmer, 1990; Altmann and Steedman, 1988) for a discussion of context in determining semantic relations.

The hierarchy of verbal concepts represents a middle ground between the surface linguistic forms and deep representations. The need for this hierarchy may be briefly illustrated by the sentences *Peter arrived in New York at 5 p.m.* and *Peter went to New York at 5 p.m.* If VM rules immediately reduce "arrived" and "went" to *ptrans-1*, then there is no way to recognize "at 5 p.m." in the first sentence as the end time of the event, and in the second sentence as the start time of the event. Hence, the VM rule for "arrived" should identify *arrive*, a subconcept of *ptrans-1*, as the verbal concept. For every thematic role in the representation of the verbal concept, its corresponding syntactic relation introducing it, and the

LTM semantic category required by the thematic role are indicated. For instance, the entry *subj* in the *ptrans-1* node says that if the concept referred to by the *subj* is subsumed by the concept *animate being* in LTM, then the *subject* is both the *actor* and the *theme*. Besides the LTM semantic categories, a user may place special restrictions on an entry. For instance, the entry for the preposition "by" in the *ptrans-1* concept (see figure 2) has a special restriction saying that the NP must not have a determiner. A general restriction encoded within the algorithm is that *a thematic role cannot be filled twice.*

In the entry for prepositions, the thematic role is followed either by the entry *strong* or *weak* meaning that the preposition suggests the thematic role strongly or weakly, respectively. (See the entries for "for" and "from" in the *action* and *ptrans-1* nodes as examples of prepositions suggesting the thematic role weakly.) In the next section, it is explained how the attachment algorithm uses these characterizations of the prepositions to attach them.

Once a verbal concept is identified by a VM rule, the syntactic relations already built by the parser are compared against the representation of the verbal concept by an algorithm that searches for the syntactic relations in the hierarchy of verbal concepts in a bottom-up fashion. Let us consider the following definition prior to explaining the algorithm. Concept B *subsumes* concept A if there is a sequence of concepts a_1, a_2, a_3 ...a_n in LTM such that a_1 = A and $a_n = B$ and the relation a_i *is-a* a_{i+1} exists in LTM for all i=1,...,n-1. If B subsumes A, then B is a superconcept of A, and A is a subconcept of B. The algorithm obeys the following rules:

1. The LTM entries in the verbal concept for an NP complement (subj, obj1, obj2, pred) and for a preposition are tried *in the order* in which they appear, and as soon as one of them subsumes the concept referred to by the PP or the NP complement, the algorithm exits with success.
2. If the syntactic relation to be compared is a NP complement, say *subj*, the algorithm searches the verbal concept and its superconcepts *until* it finds an entry called *subj*. The algorithm will return true if the LTM semantic category specified in the entry *subj* of the verbal concept *subsumes* the semantic category of the subject NP of the sentence. The algorithm will return false otherwise.
3. If the syntactic relation is a preposition, say *on*, the algorithm searches the entry for *on* in the verbal concept and its superconcepts until it finds a LTM entry for *on* that subsumes the concept of the object of the preposition, in which case the algorithm returns true and exits, or until it finds the root node of the hierarchy, in which case the algorithm returns false.

The NP complements of the verb and PPs are handled in a different way. The algorithm stops searching the hierarchy as soon as it finds the NP complement of the verb that it is looking for, and returns true if concept subsumption exists, otherwise it returns false. Thus, a NP complement entry in a subconcept overrides the same entry of any of its superconcepts. For instance, the entry *subj* in the verbal concept *drive* overrides the entry *subj* in the concept *ptrans-1* (see figures 2 and 3). In the case of PPs, however, the entry, say, *on* in the verbal concept *vc* does not override the same entry *on* (if one exists) in the superconcept of *vc*. Hence, the algorithm returns false if no entry is found that subsumes the concept of the object of the PP in concept *vc* or in any of the superconcepts of *vc*.

By way of example, consider the sentence *Peter left at 5 p.m. for Orlando on a train.* Let us assume that the VM rule that determines the meaning of the verb as *leave-a-place* (see figure 3) is an *end-of-clause* rule anchored in the verb. Thus, when the parser reaches the end of the sentence, all PPs are lying unattached in the structure built by the parser, and their meaning is undetermined. The first syntactic relation to be interpreted is the subject

```
(ptrans-1
  (is-a (action))
  (subj (ltm-ctgy
          (animate (actor) (theme))                ; peter sailed to delaware
          (vehicle (instrument))))                  ; the train went to tampa
  (obj1 (ltm-ctgy
          (motor-vehicle (instrument))              ; peter drove a truck to barcelona
          (unit-of-measure (distance))              ; peter walked 10 miles
          (human (accompany))))                     ; peter drove mary to tampa
  (adverb (ltm-ctgy
          (spatial-relation (to-loc))))             ; whales migrate south
  (prep
    (to (ltm-ctgy
          (physical-thing (to-loc (strong)))))      ; peter went to the sink
    (from (ltm-ctgy
          (physical-thing (from-loc (weak)))))      ; peter drove the man from
                                                    ;   spain to the theater
    (in (ltm-ctgy
          (vehicle (instrument (strong)))))         ; peter rode in an airplane
                                                    ;   to tampa
    (on (ltm-ctgy
          (vehicle (instrument (strong)))           ; peter sailed across the ocean
                                                    ;   on a boat
          (animal (instrument (strong)))))          ; peter rode on a horse to orlando
    (by (ltm-ctgy
          (vehicle                                  ; peter went by train/car/bus
            (instrument (strong
              (if% no-article))))))                 ; note that the NP of the PP
                                                    ;   should not have an article
    (with (ltm-ctgy
            (animate (accompany (strong)))          ;  peter went to the city with
                                                    ;   many friends
            (physical-thing (co-theme (weak)))))    ; peter went to the city with many suitcases
                                                    ; peter went to the city with
  )                                                 ;   many libraries
)
```

Figure 2. Ptrans-1 Primitive: It Subsumes All Verbal Concepts In Which the Actor Is also the Theme

of the sentence, "Peter." Because, the *leave-a-place* concept does not have an entry for *subj*, its superconcept *ptrans-1* is searched. There, the *subj* entry is found and LTM is searched to find out if "Peter" is subsumed by the concept *animate*. Because this is the case, "Peter" becomes the *actor* and the *theme* of the sentence, and the search ends. The next syntactic relation is the prepositional phrase "at 5 p.m." The algorithm starts searching again in the concept *leave-a-place* returning success this time, which results in making "at 5 p.m." the

```
; LEAVE-A-PLACE
;    to leave somewhere as the beginning of a ptrans
(leave-a-place
   (is-a (ptrans-1))
   (obj1 (ltm-ctgy (location (from-loc))))
   (prep
      (at (ltm-ctgy (time (init-time (strong))))              ; peter left
      (for (ltm-ctgy (location (toward-loc (strong)))))       ; peter left
)

; ARRIVE
;    to arrive somewhere at the end of a ptrans
(arrive
   (is-a (ptrans-1))
   (prep
      (in (ltm-ctgy (location (to-loc (strong)))))           ; peter arriv
      (at (ltm-ctgy (time (end-time (strong)))               ; peter arriv
                    (location (to-loc (strong)))))           ; peter arriv
   )                                                          ;   station at
)

; DRIVE
;    to operate a motor vehicle as the instrument of a ptrans
(drive
   (is-a (ptrans-1))
   (subj (ltm-ctgy (human (actor))))                          ; Peter drov
   (obj1 (ltm-ctgy (motor-vehicle (instrument))))             ; Peter drov
)
```

FIGURE 3. Subordinate Members of the Ptrans-1 Family of Verbal Concepts

init-time role. Success for the next prepositional phrase "for Orlando" is also found in *leave-a-place*, making it the *toward-loc* thematic role, not to be confused with the *to-loc*. Finally, the algorithm returns success for the PP "in a train" when searching the concept *ptrans-1*, making the PP the *instrument*.

4. ALGORITHM FOR ATTACHING PREPOSITIONS AND DETERMINING THEIR MEANING

We will now explain the main points in the algorithm, which also considers nominalizations. The detailed version of the algorithm is given at the end of this section. Let us first start with the following definitions:

(a) The verb in a clause is said to *strongly claim* a preposition if either (1) a VM rule fires and determines the meaning of the verb, or (2) the verbal concept (the meaning of the verb) has been identified and its representation, or the representation of any of its superconcepts, indicates that the verbal concept requires the preposition strongly.

(b) The verb is said to *weakly claim* a preposition if the verbal concept has been identified and its representation, or the representation of any of its superconcepts, indicates that the preposition is taken by the verbal concept weakly.

(c) The verb is said *not to claim a preposition* if the verbal concept has been identified and it does not claim the preposition either weakly or strongly.

Those nominalizations that have argument structure are essentially treated as verbs. As a consequence, they can claim a preposition strongly, or weakly, or not at all.

Nouns that claim prepositions, e.g., "ticket," "key," "trip," etc., also have representations that indicate the semantic category of the object of the preposition. For instance, "trip" is represented as:

(trip (to (ltm-ctgy (location (to-loc)))))

There is no indication in the representation about whether the noun claims this preposition weakly because, in principle, every noun may claim a preposition weakly. Thus, a point to bear in mind is that all attachments of a preposition to a noun not claiming the preposition strongly are considered as weak attachments.

The algorithm reduces to the following simple idea: attach the preposition to the closest verb, nominalization, or noun to the left of the preposition, claiming it strongly. If the verb or the nominalization claims the preposition only weakly, then save that attachment together with other possible attachments to noun phrases and consider some disambiguation heuristics. We explain this in detail by first considering the actions taken before the verb has been parsed and, then, those taken after.

4.1. Attaching PPs Preceding the Verb

There are only two types of contenders for prepositions preceding the verb: noun phrases and nominalizations. The action taken by the algorithm is simple. If a nominalization or a noun is found that claims the preposition strongly, the algorithm attaches the PP to the nominalization or the noun, determines its meaning, and exits. If the nominalization claims the preposition weakly, the algorithm saves that attachment and tries to attach the preposition to any noun phrase preceding the preposition. We explain these two actions in detail.

4.2. Nominalizations

The algorithm searches for nominalizations before attempting to attach a preposition to the verb or to other nouns. The search is performed right to left. Nominalizations are treated in a way similar to that presented in (Dahl et al., 1987). They may claim a preposition strongly or weakly, or not claim it at all, as explained below. If a nominalization is found that claims the preposition strongly, the preposition is attached to the nominalization and no other attempts to attach the preposition are made. If the nominalization takes the preposition weakly, then the possible attachment and meaning are saved, and an attempt is made to attach the preposition to the verb or other nouns.

A noun is identified as a nominalization by the slot *nom* in its lexicon entry. For example, the lexicon entry for the noun *attack* contains the slot *(nom attack-n)*, which indicates that *attack* is a nominalization whose definition is found under the atom *attack-n*. See Hull and Gomez (1996) for an analysis of verbal an noun senses of nominals and the polysemy of deverbal nominalizations. The definitions of nominalizations are similar to the definitions of verbal concepts and are stored as subconcepts of the corresponding verbal concept. In

```
(attack
   (is-a (action))
   (subj (ltm-ctgy (animate (actor)) (artifact (instrument)) (organization (actor))))
   (obj1 (ltm-ctgy (thing (theme))))
   (prep
      (with (ltm-ctgy (physical-thing (instrument (strong)))))
   ....
)

(attack-n
   (is-a (attack))
   (prep
      (of (ltm-ctgy (animate (actor (strong)))
                    (artifact (instrument (strong)))
                    (organization (actor (strong)))
                    (location (at-loc (strong)))))
      (on (ltm-ctgy (time (at-time (strong)))
                     (location (at-loc (strong)))
                     (thing (theme (strong)))))
      (by (ltm-ctgy (animate (actor (strong)))
                     (artifact (instrument (strong)))
                     (organization (actor (strong)))))
      (against (ltm-ctgy (thing (theme (strong)))))
   )
)
```

FIGURE 4. Definitions of Attack and Attack-N

our example, *attack-n* is stored as a subconcept of *attack*. Nominalizations need definitions separate from their corresponding verbal concept because, in some cases, they claim different prepositions, and in other cases, the same preposition may have different meanings under the nominalization as opposed to the verbal concept. This is true in the case of *attack*. The verbal concept *attack* does not claim the preposition *against*, while the nominalization *attack* claims it strongly. In other cases, however, both the nominalization and its corresponding verbal concept claim the same prepositions, and these prepositions *can be inherited by the nominalization from the verbal concept*. For example, the role for "with dynamite" is *instrument* for both *The terrorists attacked the embassy with dynamite* and *The terrorist attack on the embassy with dynamite shocked the public*. Sample definitions of *attack* and *attack-n* are shown in figure 4.

As an example, consider the sentence *The report tells of a daring attack on 22 March* *against a national guard unit.* The PPs "on 22 march" and "against a national guard unit" are both attached to *attack* because, according to the definition of *attack-n*, the prepositions *on* and *against* are claimed strongly.

4.3. Loop-through-Noun-Phrases Algorithm

This algorithm determines the meaning and attachment of prepositions to noun phrases. All attachments performed by this algorithm are considered to be weak attachments. Space limitations forbid us from explaining it in detail. It is complex enough to deserve a discussion of its own. The algorithm proceeds right to left, starting with the prepositional phrase whose meaning and attachment need to be determined and ending in the verb or in the rightmost thematic role that has been identified, if the verb has been parsed, or ending in the first syntactic relation of the clause, if the verb has not been parsed. Let np_1, np_2, ... np_i be noun phrases. Let *pp* be the prepositional phrase whose attachment needs to be determined. The algorithm will loop back through the noun phrases to find out which of them is modified by *pp*. The algorithm will consider the pairs np_1 *pp*, np_2 *pp*, ... np_i *pp*. Let $np - i$ *pp* be a pair under consideration. To determine if *pp* modifies np_i, the algorithm fires rules stored under the preposition of *pp*. Most of these rules access LTM, and they use the predicates *is-a* and *part-of*. For instance, for the phrase "the children on the bus with bad brakes" the rule **If concept(pp) is part-of concept(np_i) then the meaning-of-prep = part-of**, stored under the preposition "with" fires, determining its meaning and, as a consequence, the attachment of the preposition. The algorithm does not stop there but also tries to find a relation between "children" and "with bad brakes." In this case, no rule under "with" fires, resulting in only one attachment. The algorithm is essentially a knowledge-based algorithm that does not distinguish between world knowledge and semantic knowledge. However, a few general syntactic rules are incorporated in the algorithm, such as the rule that a definite noun group does not modify an indefinite noun group, e.g. in *Peter ate apples on the table*, "on the table" will not be attached to "apples."

4.4. Attaching prepositions following the verb

Once the verb has been parsed, it also becomes a contender for the preposition. The algorithm reduces to the following four rules, which are tried in the order listed. From right to left, until the verb of the clause is reached.

1. If a nominalization or a noun is found that strongly claims the preposition, (a) attach the preposition to the nominalization or noun, (b) if the verb of the current clause also claims the preposition strongly, determine the thematic role the PP stands for, and the meaning of the verb, if the verb strongly claims the preposition because a VM *prep-rule* has fired. Remember these facts, but do not behave as if the meaning of the verb has been determined, i.e., continue firing VM rules.
2. If the verb of the current clause claims the preposition strongly, attach the preposition to the verb and exit.
3. If the verb of a matrix clause claims the preposition strongly, attach the preposition to it and exit.
4. Else save all weak attachments (if any) including noun phrases attachments.

The following examples illustrate these ideas. In the sentence *Peter bought the tickets to Spain*, "to Spain" is attached to "ticket" because "ticket" claims the preposition strongly. The verb does not claim it either strongly or weakly. In the sentence, *Peter gave the donations* *to the children*, "to the children" will be attached to "donations," but because the verb also claims "to the children" strongly, the algorithm remembers the fact that there is also a verb claiming that PP strongly and that the thematic role corresponding to that PP is *beneficiary.* This impasse is resolved at the end of algorithm (see below) by a rule that undoes the attachment of the PP to the nominalization and attaches it to the verb, if the thematic role corresponding to the PP has not been assigned to the verbal concept. However, in the sentence *John gave the donations to the children to his lover*, the *recipient* role for "gave" will be filled by "to his lover," and there is no need to undo the attachment of "to the children" to "donations." In the sentence, *Peter took the tickets to Spain apart*, "to Spain" is attached to "tickets," and the fact that a VM rule has fired determining the meaning of "take" and the thematic role of "to Spain" as *to-loc* is remembered but the meaning of "take" remains unsolved. When "apart" is found, the meaning of "take" is identified as "rip-apart" and there is no need for undoing the attachment. A similar analysis is produced for *Kelly rid the office of the president of rats.*

In the sentence *Peter went to the library with Mary*, both prepositions are strongly claimed by the verb. The preposition "to" is strongly claimed by "went" because a VM rule under "to" fires, determining the meaning of *went* as *ptrans-1.* The preposition "with" is strongly taken by the *accompany* role in the representation of *ptrans-1.* In the sentence *Peter drove to Tampa in a car*, the prepositional phrase "in a car" is attached to the verb, because, in the representation of the parent node of *drive*, it is indicated that if the object of the preposition is a motor-vehicle, the PP is strongly taken by the verb (see figure 2).

In the sentence *Mary took the man she killed in the library to the basement*, the preposition "in" is only claimed by "killed," because the meaning of "took" is undetermined, and no VM rule fires. When "to" is encountered, the meaning of "took" still remains undetermined. The verb "killed" is not a contender for "to;" that is, it does not claim it strongly or weakly. When "took" is considered, a VM rule fires, resulting in determining the meaning of "took," and attaching the prepositional phrase "to the basement" to "took." In the sentence *The woman saw the man who cut the beef with a telescope*, the prepositional phrase "with a telescope" is attached to "saw" because, in the representation of "see," one of its roles, called "helping-instrument," takes "with" strongly, if the object of the preposition is glasses, telescope, etc. Had the sentence said "with a knife," the verb "cut" would have claimed the prepositional phrase.

The verb claims the preposition weakly. In the sentence *Peter ate the apples on the table*, the prepositional phrase will be attached to the verb, but also to the previous NP. The attachment to the verb is done because the *action* concept in the hierarchy of verbal concepts has an entry for "on" establishing a weak attachment to the verb (figure 1). The attachment to "the apples" is done because the Loop-through-Noun-Phrases Algorithm will fire a rule under "on" saying that "on NP" modifies a previous NP if the head noun of the NP is a physical thing. A similar situation occurs with the sentence *Peter read the books in the car.* The resulting ambiguity is resolved by accessing memory, searching for the concepts "apples on the table," and "books in the car." If those concepts are found, then the preposition is exclusively attached to that NP. If those concepts are not found in memory, the attachment to that NP is broken, and the process is repeated for any other NP (if any) claiming the preposition. In this example, because no other NP claims the preposition the process stops, resulting in attaching the prepositional phrase to the verb. If the verb of the current clause and the verb of the matrix clause both claim a PP weakly, the algorithm prefers the verb of the current clause because it ignores the verb of the matrix clause unless this verb claims the PP *strongly*, in which case it will attach the PP to the verb of the matrix clause, violating right association.

The Loop-through-Noun-Phrases Algorithm considers only those PPs placed between the rightmost thematic role that has been identified and the prepositional phrase to be attached. For instance, consider the sentence *Peter took a trip with the man with a hat.* It is reasonable to assume that the meaning of "took" in this and similar cases can be safely determined by an *obj1* VM rule that says if the object of "take" is "trip," then the verb's meaning is *ptrans-1*. Thus, the PP "with the man" will be strongly attached to the verb and identified as the *accompany* role (figure 2). Now, when the Loop-through-Noun-Phrases Algorithm tries to attach "with a hat," it does not go beyond the PP "with the man" because "with the man" is a thematic role that has been already identified. This will result in attaching "with the hat" to "the man" because this is the only remaining contender for the prepositional phrase. In general, if a constituent C has been attached to constituent D, none of the constituents between D and C claim any PP or modifier following C. This is our version of the rule of no-crossing-of-branches in tree structures.

The Verb Does Not Claim the Preposition. If neither the verb nor a nominalization in the current clause claim the preposition weakly or strongly, the only contenders are the NPs in the current clause and the verb of the matrix clause (if any). If the verb of the matrix clause claims the preposition strongly, the preposition is attached to such a verb, as indicated above. However, if the verb of the matrix clause claims the preposition only weakly, it becomes a contender for the preposition together with the NPs in the current clause. For instance, in *Mary knew that Kelly loved many books on the shelf*, neither "knew" nor "loved" claim the preposition, and "on the shelf" is attached to "many books" by the Loop-through-Noun-Phrases Algorithm. In the sentence, *Mary married the man she liked in Spain*, "liked" is not an action verb and does not claim "in Spain" strongly or weakly. Thus, "in Spain" is attached weakly to "married." However, one of our informants gets two attachments for "in Spain."

4.5. Heuristic Disambiguation Rules

When the algorithm finds a preposition that is strongly claimed by a verb or a nominalization, the preposition is immediately attached to the verb or nominalization, and the algorithm stops. Thus, no ambiguity can result from strongly claimed prepositions. On the other hand, when a preposition is weakly claimed by a verb or nominalization, the algorithm simply saves the attachment, and continues attempting to attach the preposition to other noun phrases. This may result in a list of several potential attachments, from which one needs to be selected. Some of the rules listed in step 4 of the algorithm are heuristics, mostly of a syntactic nature, which attempt to resolve the ambiguity. When all heuristics fail, the algorithm declares the sentence ambiguous and saves the list of attachments. In the tests we conducted on the MUC texts (see below), disambiguation rules were activated in about 10% of the cases. However, in the tests we performed on encyclopedic texts, they were only activated in about 5% of the cases.

4.6. End-of-Clause Steps

If the end of the clause is reached and the meaning of the verb remains unknown, the VM end-of-clause rules are tried. If none of the rules fires, interpretation cannot continue. But, if one of these rules fires determining the meaning of the verb, the thematic roles for the NP complements of the verb are identified first, and then the steps explained in the preceding subsections are applied to all PPs following the verb from left to right. There are no new elements in the end-of-clause steps, except that semantic interpretation is postponed until the very end of the clause.

However, if a conflict arose between a verb and a nominal because both of them claimed a PP strongly, then this is the time to check if the conflict has been solved. It was tentatively solved in favor of the nominal. The conflict has not been solved, (a) if the thematic role for the PP, which was remembered when the preposition was attached to the nominal, has not been assigned to the verbal concept of the logical form, or (b) the meaning of the verb remains unknown, but a VM rule fired determining the meaning of the verb and of the PP that was attached temporarily to the nominal. This action, however, was suspended. The algorithm acted as if the meaning of the verb was unknown because the PP was attached to the nominal. In both cases, the attachment of the PP to the nominal is undone and is reattached to the verb. In case (b), it is also necessary to check to see if every syntactic relation has been interpreted by applying the actions explained in the first paragraph of this subsection.

4.7. Algorithm

1. If the preposition is "of," and no VM rules for "of" have fired, attach the preposition to the preceding noun or verb. Fire the rules under "of" to determine the meaning of the preposition. Exit. (The VM rules need to be fired first because certain verbs such as "rid," "warn," "clear," etc. may have a *[of NP]* complement, e.g., *He rid the town of rats*). [2]
2. If the verb has not been parsed yet,
   (a) For each noun phrase before the preposition, right to left, if a nominalization or a noun is found that *strongly claims* the preposition, attach the preposition to the nominalization or noun, save its meaning, and exit. Else, if the nominalization *weakly claims* the preposition, save the attachment to the nominalization and the corresponding meaning of the preposition.
   (b) Try to attach the preposition to the previous noun phrases which are not nominalizations by determining the possible meanings of the preposition (Loop-through-Noun-Phrases Algorithm). Save the attachments and meanings. Note that attachments performed by the Loop-through-Noun-Phrases Algorithm are always considered to be weak attachments.
   (c) If a single attachment has been identified, exit. Else, read step 4.
3. If the verb has been parsed,
   (a) For each noun phrase before the preposition and after the verb, right to left, if a nominalization or a noun is found that *strongly claims* the preposition, then
      i. attach the preposition to the nominalization or noun.
      ii. If the verb of the current clause also claims the preposition strongly, determine the thematic role the PP stands for and the meaning of the verb, if a VM rule has fired. Remember these facts.
      iii. Exit.

      Else, if the nominalization *weakly claims* the preposition, save the attachment to the nominalization and the corresponding meaning of the preposition.
   (b) If the verb *strongly claims* the preposition, attach the preposition to the verb, save its meaning, and exit. This will override any previous weak attachments.

[2] We are indebted to an anonymous referee for this observation.

(c) If the verb of a matrix clause, if any, *strongly claims* the preposition, and if the meaning of the verb of the current clause has already been determined, attach the preposition to the closest such verb, save its meaning, and exit. Again, this overrides any previous weak attachments.

(d) If the verb *weakly claims* the preposition,

   i. Save the attachment to the verb and the meaning of the preposition.

   ii. Right to left, try to attach the preposition to the noun phrases placed between the rightmost thematic role that has been identified and the preposition, by determining the possible meanings of the preposition (Loop-through-Noun-Phrases Algorithm).

   iii. If a single attachment has been identified, exit. Else, read step 4.

(e) If the verb does not claim the preposition,

   i. Right to left, try to attach the preposition to the noun phrases placed between the rightmost thematic role that has been identified and the preposition, by determining the possible meanings of the preposition (Loop-through-Noun-Phrases Algorithm).

   ii. If a single attachment was identified, exit.

   iii. If no attachment was identified in the current clause, and there is a matrix clause whose verb *weakly claims* the preposition, attach the preposition to the closest such verb, save its meaning, and exit.

4. If more than one possible weak attachment has been identified, attempt to resolve the ambiguity by using the rules 4(a) to 4(e) explained below, and by checking for the existence in memory of entities denoted by definite noun groups. These rules attempt to select one attachment out of the list of weak attachments determined by the algorithm.

   (a) If the prepositional phrase is weakly attached to the verb and to a noun phrase with zero determiners, prefer the attachment to the verb.

   (b) If the prepositional phrase is weakly attached to the verb and to a noun phrase, prefer the attachment to the noun phrase if the noun phrase is a definite reference and the concept denoted by NP PP exists in memory. Otherwise, prefer the attachment to the verb.

   (c) If the prepositional phrase has been attached to noun phrases only, prefer the rightmost noun phrase.

   (d) If the prepositional phrase is weakly attached to the verb and the object of the preposition is a location or time, prefer the attachment to the verb.

   (e) Save the multiple attachments and exit.

4.8. End-of-Clause Steps

1. If there is a verbal concept claiming a PP that has been attached to a nominalization or a noun strongly and that thematic role has not been assigned to that verbal concept, undo the attachment of the PP to the nominalization or noun and attach it to the verbal concept. Exit.

2. If the verbal concept has not been determined,

   (a) If a PP has been attached to a nominalization or noun, and a VM rule fired for that PP, undo the attachment of the PP to the noun or nominalization and attach it to the verb.

  (b) Else, fire the VM *end-of-clause* rules in order to determine the meaning of the verb. If the rules do not succeed, interpretation cannot continue.

3. If the verbal concept has been determined,
   (a) For each NP complement, identify its thematic role using the representation of the verbal concept.
   (b) For each unattached preposition after the verb (left to right), apply step 3 of the main algorithm and all of its substeps.
   (c) Apply step 4 of the main algorithm, if needed.

5. DISCUSSION

How is it decided if a verbal concept in the hierarchy claims a preposition strongly or weakly? This is an empirical question. If, in most situations, a verb suggests a preposition strongly, then it is safe to represent it that way, because the algorithm can go wrong only in very few cases. Moreover, these cases can be handled by exception rules. However, even if a preposition is represented as weakly claimed by a verb, this may not result in an ambiguous situation, because the verb may be the only contender for the preposition, as in the sentence discussed above, *Mary married the man she liked in Spain.*

Does our algorithm produce ambiguous cases? Certainly. For the sentence *Peter ate the apples on the table*, the algorithm will attach “on the table” to the verb and to “the apples,” unless there is a reference in memory to “apples on the table,” in which case the algorithm will prefer the attachment of the PP to the “apples.” Of course, one may decide to write an entry for “on” in the verbal concept *eat* saying that, if the LTM category of the NP is *is-a table*, then the preposition is strongly taken by the verb. In that case, the algorithm will produce only the attachment to the verb. That has not been our choice, but quite a few of our informants are consistent with that approach, since tables are usual places to eat.

What if a concept in the sentence is not subsumed by the LTM category specified for the preposition in the verbal concept? For instance, let us assume that the verbal concept *eat* has at least two entries for the preposition “with:” one saying that if the object of the preposition is *is-a eating-utensil* in LTM, then the preposition is strongly claimed by the verbal concept, and the thematic role of the PP is *instrument*; and the other one saying that if the object of the preposition is *is-a animate*, then the PP is strongly claimed by the verb, and the thematic role is *accompany*. Now, if the algorithm reads the sentences *Peter ate the cake with a fork* and *Peter ate the cake with a Frenchman* without knowing that forks are eating utensils and that Frenchmen are animate beings, then “a fork” and “a Frenchman” will be attached to the verb and also to “the cake.” A similar situation will occur if the algorithm reads the sentence *Peter ate the cake with the cherry* without knowing that cherries are food. The interpreter keeps the multiple interpretations until the system reads something that allows it to choose one. In the example above, that will be the case if the system reads that Frenchmen are humans, or that many Frenchmen eat cheese. It is well beyond the limits of this paper to discuss this in detail.

Because our method depends on having hand-crafted lexical entries, one may ask how easy is the task of building these lexical entries beyond narrow domains? One may think that the method has little application beyond limited domains because of the burden of constructing the lexical entries. The VM rules, however, are very easy to construct. It is a manual process but one that can be done very fast, similar to the construction of syntactic subcategorizations for verbs. In the initial stages of the construction of the hierarchy of verbal concepts, the addition of new elements required considerable analysis. But now that we have built the basic framework, the addition of new elements is a rather easy task. The hierarchy of action verbs facilitates the insertion of new items because, in many cases, most of the thematic roles are inherited from superconcepts. In other cases, the construction of a lexical entry, reduces, for the most part, to cutting and pasting entries in the existent lexical items. In the winter of 1993, we used the semantic interpreter for the task of acquiring knowledge about the diets of animals from unedited articles of the *The World Book Encyclopedia*(World Book, Inc., 1994). This required not only the interpretation of sentences with verbs subsumed by *ingest*, but also sentences with many other verbs unrelated to *ingest*, but from which an *ingest* relation may be inferred. For instance, it became necessary to interpret sentences such as *A bear uses its claws to dig up roots, ants, and termites* or *Bears are fond of honey and will rip apart beehives to get it*, because one can infer that bears eat roots and honey. Most of the action verbs and VM rules were built by a graduate student and two undergraduate students. The hierarchy of the verbs for this application can be found in (Gomez et al., 1994). A related problem is the construction of the ontological categories for content words. In our encyclopedic application, this proved to be more of a problem, requiring more attention on the part of the leader of the project than the construction of the lexical entries for verbs. Having the ontological categories consistently and generally defined across domains is essential to our method, because most VM rules and all verbal concepts access the ontological categories of the syntactic relations. For instance, consider the VM object-rule for "take" explained above, **R2. If concept(np) is-a LTM(medicament) then meaning-of-verb = ingest.** For this rule to work for "pill," "tablet," "laxative," "antihistamine," etc., it is necessary that the lexicon be defined consistently.

Recently, we have begun studying WordNet (Miller, 1993), which is an on-line public domain tool. WordNet organizes lexical knowledge in terms of word senses, which are captured in a reduced set of ontological primitives organized into an inheritance hierarchy, similar to the ones proposed in this work. There are striking similarities between some of the ontological categories in WordNet and the ones that we have been using in our system for some eight years now, and there are also important differences, stemming from the different criteria used in designing the ontology. There are also some problems with using WordNet for understanding in its present form: for example, the underspecification of some of its concepts, no ontological categories for adjectives and adverbs, and others. But, in general, we are highly impressed by WordNet, and are presently mapping WordNet ontological categories into ours. Our aim is that a somewhat modified WordNet lexical knowledge base will become the lexicon of our system. We are also studying the possibility of connecting the verb senses in WordNet to the ontology by means of our VM rules, and introducing the VM rules in the definition of content words. The aim is to overcome the static nature of the senses listed in WordNet by connecting them to the semantic interpretation process. If this effort succeeds, we will be ready to perform semantic interpretation on a large scale.

6. TESTING

To verify the effectiveness of our approach, we analyzed how well the algorithm performed three tasks: determining the meaning of the verb, determining the attachment of prepositions, and determining the meaning of prepositional phrases. The MUC-3 texts were selected because they are well known and easily accessible, besides providing a varied and realistic source of sentences. A subset of 100 texts were selected for our testing purpose. Because our approach is verb-centered, we needed first to decide which verbs we were going to test. In order to avoid any bias in the selection of the test verbs, a word frequency histogram was used to select the 48 most frequent verbs found in the 100 MUC messages used

TABLE 1. 48 Most Frequent Verbs in the first 100 texts of the MUC-3 test set.

SAY	REPORT	KILL	FIND
ADD	TAKE	BELIEVE	MAKE
INVOLVE	MURDER	KNOW	CARRY
DESTROY	KIDNAP	STATE	RECEIVE
INJURE	LOCATE	ATTACK	WOUND
USE	CLAIM	ANNOUNCE	PARTICIPATE
CALL	TELL	ARREST	ARRIVE
DENY	LAUNCH	LINK	PUNISH
PERPETRATE	ISSUE	HOLD	OCCUR
ENTER	DIE	ALLEGE	CAUSE
ASK	FORCE	RESULT	CONSIDER
TRY	CONDUCT	SEND	INVESTIGATE

for testing. This resulted in a list of verbs that included many that were domain independent as well as some that were related to the MUC-3 domain (see Table 1). This should come as no surprise because the most frequent verbs are typically those that have many uses, such as “take” “make”, “hold”, etc.

For each verb on the list, we extracted every sentence in the 100 MUC-3 texts that contained a form of that verb. This resulted in a testing corpora of 680 sentences from the 1000 sentences that make up the 100 MUC-3 texts. Some sentences were selected more than once because they contained more than one frequent verb.

Once the verbs and their corresponding sentences had been selected, the next step was to define the rules that determine the meaning of the verb, the verbal concepts and their place in the hierarchy, and the concepts to be added to the ontology. Because most of the 48 verbs selected already had some definition within our system, few new definitions had to be added. New definitions were needed, however, for *deny, perpetrate, kidnap, launch, link, allege, wound, participate*, and *punish.* The members of the testing team, which included beginner graduate students and some undergraduate students, made sure to construct the new definitions independent of the testing corpora so that the definitions would not be skewed to handle only those texts.

6.1. Testing Method

The testing procedure used was the following. A sentence for the verb under consideration was selected, and the algorithm was given the sentence containing that verb. Spanish proper names were not recognized, but instead were given as hyphenated expressions. Also, because this paper is not concerned with the handling of coordinate conjunctions or appositions, their occurrences in the text were ignored. The output of the algorithm for the sentence was checked to determine the correctness of the system-generated verb meaning, prepositional attachment, and prepositional meaning. If the attachment of a prepositional phrase is incorrect, the meaning will also be incorrect. Therefore, we only analyzed the system-determined prepositional meanings for those prepositions that were correctly attached. Successes and failures were noted, and then the next sentence for the current verb was selected and this process was repeated.

Table 2. Test Results

	MUC-3	World Book
Verb Meaning Correctly Determined	93.7%	98.8%
Verb Meaning Incorrectly Determined	2.8%	0.6%
No Verb Meaning Found	3.5%	0.6%
PP Correctly Attached	88.1%	98.3%
PP Incorrectly Attached	1.3%	1.4%
No Attachment Found	2.6%	0.3%
Ambiguous Attachment	8.0%	0.0%
PP Meaning Correctly Determined	81.3%	89.7%
PP Meaning Incorrectly Determined	2.9%	0.3%
No Meaning Found	15.8%	10.0%

6.2. Test Results

The test results are shown in Table 2. The algorithm selected the correct meaning for the verb 93.7% of the time. An incorrect meaning was a consequence of rules that were too general or of an *end-of-clause* rule failure that occurred because the LTM semantic categories of the verb rules did not succeed and the *end-of-clause* rules were applied in the wrong situation. Failure to determine the meaning of the verb occurred in situations that were not handled by the existing rules, i.e., failures due to omission, or because we chose not to include an *end-of-clause rule.* For the 48 verbs selected, the data shows that the algorithm exhibited a high rate of success, with mistakes being evenly split between either incorrect meanings or failures to find any meaning.

The results for prepositional attachment are listed in the second section of Table 2. The algorithm attached correctly 88.1% of the PPs. Because the attachment and meaning determination of prepositions is driven by the verbal concept, in most cases the attachment of a preposition results in determining its meaning too. The percentages in the table are computed based on the total number of prepositional phrases in the text. Failures that led to incorrect attachments were due to missing LTM entries for the verbal concepts, entries that were too specific, or entries that were too general.

The meaning of the prepositional phrase was correctly determined in 81.3% of the cases as shown in the bottom section of Table 2. PPs for which no meaning could be found were much more frequent than incorrect meanings. In many cases, the meaning of some prepositions could not be found in those situations in which the object of the preposition was not categorized in the ontology. For instance, the meaning of the PP "in detail" in the sentence *Dr. Julio Alfredo Samayoa reported today in detail on the murder of Msgr Oscar Arnulfo Romero* could not be determined because "detail" was not categorized in LTM.

These algorithms have also received intense testing, which has been highly positive, on the acquisition of knowledge about the diet and habitat of animals from unedited articles taken from *The World Book Encyclopedia*(World Book, Inc., 1994) mentioned in Section 5. In Gomez et al. (1994) we reported an evaluation of the system regarding its performance in the overall task of knowledge acquisition. In March of 1995, we conducted a formal testing of this algorithm in the encyclopedic articles about animals. Instead of defining new verbal

TABLE 3. Verbs used in selecting sentences for the World Book test set.

BITE	BREED	CAPTURE	CATCH
CHASE	CHEW	CONSUME	DAMAGE
DEVOUR	DIG	DIGEST	DRINK
DWELL	EAT	FEED	FIND
FORAGE	GORGE	GULP	HUNT
INHABIT	LIVE	MIGRATE	NEST
NIBBLE	PREFER	PREY	RANGE
REPRODUCE	ROAM	SCOOP	SEARCH
SEIZE	SPEND	STALK	STEAL
SWALLOW	TAKE		

concepts and VM rules, we selected 38 of the most frequent verbs for which we had already built VM rules and verbal concepts (see Table 3). Most of these verbs trigger verbal concepts related to *inhabit* and *ingest*, or verbal concepts from which those verbal concepts can be inferred.

The testing proceeded by randomly selecting 20 sentences from the encyclopedic articles about animals, containing at least one of the verbs in Table 3. The test results are depicted in Table 2. The improvement over the MUC texts may be due to the following reasons: The sentences in the World Book Encyclopedia are considerably shorter (the average length of the sentences in the animal articles is 14.2 words/sentence) than those in the MUC texts (average length is 24.9 words/sentence). The encyclopedic articles are carefully written, while some MUC texts seem to be bad translations of Spanish texts. Finally, when we undertook the encyclopedic task, we had already constructed the basic framework for verbal concepts and VM rules resulting not only in an easier task, but also in one with better results.

7. RELATED RESEARCH

Brill and Resnik (Brill and Resnik, 1994) present a method for PP attachment that uses superficial knowledge extracted from a corpus. Their method consists of entering the text into an initial-state annotator which uses right association (Kimball, 73). The initially annotated text is compared to a manually annotated corpus. As a result of this comparison, transformations are learned that can become part of the initial annotator. For instance, three instances of transformations learned by the method are: Change attachment of PP from NP to VP if preposition is “at,” “as,” or “by.” One of the limitations of their method is that they apply it only to structures of the form V N1 p N2, where *V* is the verb, *N1* is the head of its object NP, *p* is the preposition, and *N2* is the head noun of the object of the preposition. Another problem is that their method says nothing about determining the meaning of the preposition, something that is essential in the method we propose. However, one of its advantages is that it is not semantic-intensive, like ours. However, when they incorporated some semantic information in their method as provided by WordNet (Miller, 1993) they increased their accuracy to 81.8%. The same comments apply to a similar method reported by (Ratnaparki and Roukos, 1994). This is a statistical method that uses identities of words and word classes. The latter are not manually constructed, but gathered by clustering mutual information in a training corpus (Brown et al., 1992). They attach the preposition to the item (V or N1) that has the highest conditional probability. They report an accuracy of 81.8%. Both methods use the Penn Treebank's Wall Street Journal Corpus for training and test data.

There is a similarity between our decision to leave PPs unattached and some of the ideas in description theory (Marcus et al., 1983; Marcus, 1987; Marcus and Hindle 1990). Parsers based on description theory (D-theory) are descendants of Marcus' PARSIFAL deterministic parser (Marcus, 1980), but they do without lookahead by adopting a representation of syntactic structure that underspecifies the dominance relations between syntactic relations. According to D-Theory, the output of a natural language parser is not the construction of a *tree* with nodes directly dominating other nodes, but rather a structure indicating that node x dominates node y, rather than x directly or immediately dominates y. This type of *minimal commitment models* (Weinberg 1993; Gorrell 1991) is very much in tune with our approach, *in so far as the immediate dominance relations between two syntactic relations are computed mostly on a semantic basis.* The motivation behind D-theory, however, is Marcus' determinism hypothesis that says that natural language can be parsed by a deterministic mechanism; that is, a mechanism that constructs the parse structures irrevocably. Because modifier attachment cannot be done irrevocably on the basis of syntax by a deterministic mechanism, it needs to be left underspecified. Not even object NPs can be determined irrevocably, i.e., *I drove my aunt from Peoria's car* (Marcus et al., 1983). As a consequence, the structure that our parser builds is slightly less impoverished than that built by a D-theory parser, because our parser recognizes NP complements and *time* NP verb adjuncts. The latter are recognized on the basis of some semantic features, but not always satisfactorily in unrestricted texts. However, a solution more consistent with the approach taken here is to let the semantic interpreter recognize *time* NP verb adjuncts. (For a tentative solution of this problem, see (Gomez, 1995)).

Recently, Pritchett has proposed a strong syntactic model of attachment (Pritchett, 1992). His main idea is expressed in the following principle:

> *Generalized Theta Attachment(GTA): Every principle of syntax attempts to be maximally satisfied at every point during processing.*

where "every principle of syntax" refers to the principles of Government and Binding theory (Chomsky, 1981). For instance, the PP "to the children" in *Peter gave the tickets to the children* will be attached as the second argument of "give," rather than as a modifier of "tickets." GTA makes this attachment decision because the argument attachment maximally satisfies the Theta Criterion (Chomsky, 1981), which informally says that each thematic role must be assigned to one argument and each argument must have one thematic role. Note that the theta criterion and the theta attachment do not have access to the semantic content of the thematic roles, but only to the argument structure of the verb. Thus, attachment decisions are based solely on syntax. The advantages of GTA over minimal attachment, however, are that it is not oblivious to the argument structure of the verb and that it derives the attachment decisions from a syntactic theory. But like other syntax-based methods, GTA determines the *final* attachment of the PP wrongly or makes no predictions. Consider the following example:

```
(1) (a) Peter gave the tickets to the children.
    (b) Peter gave the tickets to Spain to the children.
```

It is unclear how GTA will behave in (1.b). If GTA recognizes an argument position for "tickets," then it needs to resolve the conflict of both "gave" and "tickets" claiming "to

Spain." If it resolves the conflict in favor of "tickets" then everything works very nicely in (1.b), but very badly in (1.a), and if the conflict is resolved in favor of the verb, then things look very bad in (1.b). Let us consider the following two additional examples.

```
(2) a. Peter repaired the table with the broken leg.
    b. Peter repaired the table with the screwdriver.

(3) a. Peter ate a cake with a cherry.
    b. Peter ate a cake with a spoon.
```

Let us assume that "repair" assigns two thematic roles: *theme* and *instrument*. GTA will attach "with the broken leg" and "with the screwdriver" to the verb, which is clearly not the final attachment in (2.a). If one decides that "repair" assigns only *theme*, then GTA will not predict any preference attachment for any of the PPs. The same comments apply to example (3) *mutatis mutandis*. Pritchett recognizes the problem with *instrument*, and he calls it a "quasi-argument." We presume that *accompany* presents similar problems to those for *instrument*. The difficulty with locative and temporal adjuncts is even more serious for GTA, as it is generally recognized. In our approach, we are also aware of the quasi-argument status of some roles, and we have recognized it by categorizing the prepositions of some verbal concepts as weakly required by them. But, the final decision in our method lies in accessing the semantic content of the thematic roles. In conclusion, GTA is a principle that is insufficient to determine the final attachment of PPs. However, one may say that GTA or, for that matter, other syntactic structural parsing preferences, are not concerned with *determining the final reading of a sentence, but with the initial, on-line preferences.*

In our model, knowledge is represented declaratively in frame-like structures. In (Sondheimer et al., 1984), frame-like structures, KL-ONE structures in fact, are also used to guide semantic interpretation in an application domain. However, other than for using frame-like structures, the organization of knowledge presented here and the overall approach to the interpretation task clearly differ from those presented in that work.

Is this approach a descendant of conceptual parsers (Birnbaum and Selfridge, 1981)? One may think so, and some people have. However, parsing in our system is done by a syntax-based algorithm. The parsing of a sentence may succeed, and no interpretation may be produced for it. That will be the case whenever the meaning of the verb remains unknown. In conceptual parsers, the semantic component drives the entire processing of the sentences. In some versions (Lytinen, 1991), the semantic processing is performed first, and then the syntactic component is searched to validate the interpretation syntactically. In our approach, the parser proposes, and the semantic interpreter decides. But, what our parser proposes is not much. For instance, in the case of PPs, our parser proposes nothing in terms of attachment. Likewise, the complements of the verb built by the parser are just clues used by the semantic interpreter to build the thematic roles. Could the parser be overridden by the semantic interpreter once the meaning of the verb has been determined? In this case, the semantic interpreter could rely on a simpler processor such as a finite state analyzer that would be instructed to segment the words of the sentence into NPs, PPs, etc. by constantly interacting with the semantic interpreter. If this were possible, then the approach presented here could become a descendant of conceptual parsers.

Of the semantic-based approaches (Dahlgren, 1986) to prepositional attachment, the closest to ours is (Wilks et al., 1987). Both approaches take into consideration the arguments of both nouns and verbs in attaching PPs. Also, they go beyond the lexical aspect of verbs in attaching prepositions and rely on a deep semantic representation of verbs in which not only the prepositions required by the verbs are represented, but also semantic and world knowledge of the object of the prepositional phrase. However, our representation and organization of knowledge are centered in the hierarchy of verbal concepts and LTM concepts, and the characterization of prepositions as weakly or strongly suggested by the verb clearly differs from Wilks' approach. Another significant difference is that our algorithm integrates the attachment of prepositions and the determination of the meaning of verbs. This is important since semantic approaches rely on the meaning of the verb to attach prepositions. Another main difference lies in the overall way in which semantics and syntax are represented and combined in producing the interpretation of the sentences. In Hirst's (Hirst, 1984), definite noun phrases (those introduced by the article "the") are strongly preferred in determining the attachment of prepositions. This aspect has been incorporated in our algorithm. However, definite noun phrases are only considered by our algorithm when the verbal concepts claim the preposition weakly and a decision needs to be made whether to attach the PP to the verb or to a NP.

8. CONCLUSIONS

This paper has presented a general algorithm for semantic interpretation that hinges on (a) lexical rules that use the syntactic realizations of words and their semantic category to determine the meaning of the verb, called the verbal concept, and the attachment and meaning of PPs, (b) the representation of verbal concepts in a classification hierarchy, (c) characterizing prepositions as weakly or strongly required by the verb, and (d) an ontology for content words. We have emphasized the determination of the attachment and meaning of the prepositions, because our parser does not attach prepositional phrases, and also because determining the attachment and meaning of prepositional phrases is a necessary condition for understanding to occur. The algorithm relies upon a shallow parse and delays commitment to all semantically important decisions until the meaning of the verb is determined. Once the meaning of the verb is determined, procrastination ends and semantic interpretation shifts from neutral to high gear. These techniques are being intensively used as the front end of SNOWY (Gomez and Segami, 1991; Gomez and Segami, 1989; Gomez et al., 1994), a system for the acquisition of knowledge from expository texts.

Acknowledgments

We are grateful to Donna Markey and Rebecca Parsons for reading several drafts of this manuscript, to Bruce Martin for assisting with the testing of the algorithm, and to three anonymous referees whose incisive comments have improved the quality of this paper greatly.

This research is being funded by NASA-KSC Contract NAG-10-0058.

REFERENCES

ALTMANN, G . and STEEDMAN, M. Interaction with Context During Human Sentence Processing. Cognition, **30**:191-238, 1988.

BIRNBAUM, L., and M. SELFRIDGE. 1981. Conceptual Analysis of Natural Language. Inside Computer Understanding, R. Schank and C. Riesbeck (eds.), Lawrence Erlbaum, Hillsdale, NJ.

BRILL, E., and P. RESNIK. 1994. A Rule-Based Approach to PP Attachment Disambiguation. *In* Proceedings of COLING-94. Kyoto, Japan.

Brown, Della Pietra, deSouza, Lai and Mercer, 1992. Class-Based n-Gram Models of Natural Language. *Computational Linguistics*, **18**(4):467-480.

CARDIE, C., and W. LEHNERT. 1991. A Cognitively Plausible Approach to Understanding Complex Syntax. *In* Proceedings of the Ninth National Conference on Artificial Intelligence, Anaheim, CA, pp. 117-124.

CHOMSKY, N. 1981. Lectures on Government and Binding. Foris, Dordrecht, The Netherlands.

DAHL, D.A., M.S. PALMER, and R.J. PASSONNEAU. 1987. Nominalizations in PUNDIT. *In* Proceedings of the 25th Annual Meeting of the ACL, pp. 131-137.

DAHLGREN, K., and J. MCDOWELL. 1986. Using Commonsense Knowledge to Disambiguate Prepositional Phrase Modifiers. *In* Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence, Philadelphia, PA, pp. 589-593.

FORD, M., J. BRESNAN, and R. KAPLAN. 1981. A Competence-Based Theory of Syntactic Closure. The Mental Representation of Grammatical Relations, J. Bresnan (ed.), MIT Press, Cambridge, MA.

FRAZIER, L., and J. FODOR. 1978. The Sausage Machine: A New Two-stage Parsing Model. Cognition, **6**:291-325.

GIBSON, E., and N. PEARLMUTTER. 1993. A Corpus-Based Analysis of Constraints on PP Attachment to NPs, Manuscript, Carnegie Mellon University.

GOMEZ, F. 1989. WUP: A Parser Based on Word Usage. Technical Report UCF-Tech-89-2, Department of Computer Science, University of Central Florida.

GOMEZ, F. 1995. Ontological Categories, Polysemy and Verb Adjuncts. Technical Report UCF-Tech-95-2. Department of Computer Science, University of Central Florida.

GOMEZ, F., and C. SEGAMI. 1989. The Recognition and Integration of Concepts in Understanding Scientific Texts. Journal of Experimental and Theoretical Artificial Intelligence, **1**:51-77.

GOMEZ, F., and C. SEGAMI. 1991. Classification-Based Reasoning. IEEE Transactions on Systems, Man and Cybernetics, **21**:644-659.

GOMEZ, F., R. HULL, and C. SEGAMI. 1994. Acquiring Knowledge from Encyclopedic Texts. *In* Proceedings of the ACL's 4th Conference on Applied Natural Language Processing, ANLP94, Stuttgart, Germany, pp. 84-90.

GORRELL, P. 1991. Subcategorization and Sentence Processing. In Berwick, R.C., Abney, S. P. and Tenny, C. (eds.) Principle-Based Parsing: Computation and Psycholinguistics. Kluwer Academic, Dordrecht, The Netherlands.

GRISHMAN, R., J. STERLING, and C. MACLEOD. 1991. New York University: Description of the PROTEUS System as Used for MUC-3. *In* Proceedings of the Third Message Understanding Conference (MUC-3), San Diego, CA, pp. 183-190.

HINDLE, D., and M. ROOTH. 1991. Structural Ambiguity and Lexical Relations. *In* Proceedings of the 29th Conference of the Association for Computational Linguistics, pp. 229-236.

HIRST, G. 1984. A Semantic Process for Syntactic Disambiguation. *In* Proceedings of the National Conference on Artificial Intelligence, Austin, TX, pp. 148-152.

HOBBS, J. R. 1991. SRI International: Description of the TACITUS System as Used for MUC-3. *In* Proceedings of the Third Message Understanding Conference (MUC-3), San Diego, CA, pp. 200-206.

HOBBS, J. R., and J. BEAR. 1990. Two Principles of Parse Preference. *In* Proceedings of the 13th Conference on Computational Linguistics, pp. 162-167.

HULL, R., and F. GOMEZ. 1996. Semantic interpretation of nominalizations. *In* Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland, OR, pp. 1062-1068.

KIMBAL, J. 1973. Seven Principles of Surface Structure Parsing in Natural Language. Cognition, **2**:15-47.

LYTINEN, S. L. 1991. Semantics-first Natural Language Processing. *In* Proceedings of the Ninth National Conference on Artificial Intelligence, Menlo Park, CA, pp. 111-116.

MCROY, S. W. 1992. Using Multiple Knowledge Sources for Word Sense Discrimination. Computational Linguistics, **18**:1-30.

MARCUS, M. 1980. A Theory of Syntactic Recognition of Natural Language. MIT Press, Cambridge, MA.

MARCUS, M. 1987. Deterministic Parsing and Description Theory. In P. Whitelock, M. Wood, H. Somers, R. Johnson, and P. Bennett (eds.), *Linguistic Theory and Computer Applications.* Academic Press.

MARCUS, M., and D. HINDLE, 1990. "Description Theory and Intonation Boundaries" In G. T. Altmann (ed.) Cognitive Models of Speech Processing. MIT Press.

MARCUS M., D. HINDLE, and M. FLECK, 1983. D-Theory: Talking about talking about trees. *In* Proceedings of the Association for Computational Linguistics, MIT, June 1983, pp. 129-136.

Miller, G. A. 1993. Nouns in WordNet: A Lexical Inheritance System. In "Introduction to WordNet: An On-line Lexical Database", a revised version of "WordNet: An on-line lexical database", International Journal of Lexicography, **3**(4), pp. 235-312, 1990.

Palmer, M. S. 1990. *Semantic Processing for Finite Domains.* Cambridge University Press.

Pritchett, B. L. 1992 Grammatical Competence and Parsing Performance. The University of Chicago Press.

Ratnaparki, A., and S. Roukos, 1994. A Maximum Entropy Model for Prepositional Phrase Attachment. *In* Proceedings of the ARPA Workshop on Human Language Technology. Plainsboro, NJ, pp. 250-255.

Rich, E., J. Barnett, K. Wittenburg, and D. Wroblewski. 1987. Ambiguity Procrastination. *In* Proceedings of the Sixth National Conference on Artificial Intelligence, Seattle, WA, pp. 571-574.

Riesbeck, C. K., and C. E. Martin. 1986. Direct Memory Access Parsing. Experience, Memory and Reasoning, J. Kolodner and C. K. Riesbeck (eds.), Lawrence Erlbaum Associates, Hillsdale, N.J.

Schank, R. 1975. Conceptual Information Processing. American Elsevier, New York, 374 pages.

Schubert, L. K. 1986. Are There Preference Tradeoffs in Attachment Decisions. *In* Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelphia, PA, pp. 601-606.

Slator, B. M., and Y. Wilks. 1991. PREMO: Parsing by Conspicuous Lexical Consumption. Current Issues in Parsing Technology, M. Tomita (ed.), Kluwer Academic Publishers, Boston, MA, pp. 85-102.

Sondheimer, N. K., R. M. Weischedel, and R. J. Bobrow. 1984. Semantic Interpretation Using KL-ONE. *In* Proceedings of the Eleventh International Conference on Computational Linguistics, pp. 101-107.

Tomita, M. 1985. Efficient Parsing for Natural Language. Kluwer Academic Publishers, Boston, MA.

Weinberg, A. 1993. Parameters in the Theory of Sentence Processing: Minimal Commitment Theory Goes East. Journal of Psycholinguistic Research, **22**(3): 339-364.

Whittemore, G., and K. Ferrara. 1990. Empirical Study of Predictive Powers of Simple Attachment Schemes for Post-Modifier Prepositional Phrases. *In* Proceedings of the 28th Meeting of the Association for Computational Linguistics, pp. 23-30.

Wilks, Y. 1975. A preferential Pattern-Seeking Semantics for Natural Language Inference. Artificial Intelligence, **6**:53-74.

Wilks, Y., X. Huang, and D. Fass. 1987. Syntax, Preference and Right Attachment. *In* Proceedings of the Tenth International Joint Conference on Artificial Intelligence, Milano, Italy, pp. 779-784.

Wilks, Y., D. Fass, C. Guo, J. McDonald, and T. Plate. 1989. A Tractable Machine Dictionary as a Resource for Computational Semantics. Computational Lexicography for Natural Language Processing, B. Boguraev and T. Briscoe, (eds.), Longman, Harlow, Essex, England.

Appendix

Sample Session

This annotated session shows how the interpretation of a sentence proceeds, in particular, verb meaning determination, and attachment and meaning of prepositions. Each time the parser parses a new syntactic relation, it calls the routine ``interpret'' to perform the semantic analysis.

Example 1.

```
>>> peter took the train to tampa.

 0: (interpret g4946 =str subj ((pn peter)))

    g4946
    (subj ((pn peter)))

 0: returned
    g4946
    (subj ((parse ((pn peter))) (interp (peter))))
```

Remarks: ``interpret'' is called to interpret the ``subj'' in the structure g4946. The original ``subj'' slot is saved in the ``parse'' subslot, and an ``interp'' slot is added containing the interpretation of the noun phrase.

```
 0: (interpret g4946 =str verb ((main-verb take took) (tense sp)))

    g4946
    (subj ((parse ((pn peter))) (interp (peter)))
     verb ((main-verb take took) (tense sp)))

   1: (rule-eval nil)
   1: returned nil

 0: returned
    g4946
    (subj ((parse ((pn peter))) (interp (peter)))
     verb ((main-verb take took) (tense sp)))
```

Remarks: ``interpret'' is called to interpret the ``verb'' of the clause g4946. Since no VM verb-rules are found under ``took,'' the meaning of the verb cannot be determined, and the original clause is returned unchanged.

```
 0: (interpret g4946 =str obj1 ((dfart the) (noun train)))

    g4946
    (subj ((parse ((pn peter))) (interp (peter)))
     verb ((main-verb take took) (tense sp))
     obj1 ((dfart the) (noun train)))

   1: (rule-eval (fire-one
                  ((if% is-a obj1 bath) (meaning-of-verb-is bathe)
                   (thematic-role-is none))
                  ((if% is-a obj1 trip) (meaning-of-verb-is ptrans-1)
                   (thematic-role-is none))))
   1: returned nil

 0: returned
    g4946
    (subj ((parse ((pn peter))) (interp (peter)))
     verb ((main-verb take took) (tense sp))
     obj1 ((parse ((dfart the) (noun train))) (ref (definite))
          (plural nil) (interp (train))))

Remarks: when ``obj1'' is found, the interpreter retrieves the VM
obj1-rules under ``took.'' None of them fire in this case, so control
is returned to the parser to read the next syntactic relation.


 0: (interpret g4946 =str prep (to ((pn tampa))))

    g4946
    (subj ((parse ((pn peter))) (interp (peter)))
     verb ((main-verb take took) (tense sp))
     obj1 ((parse ((dfart the) (noun train))) (ref (definite))
          (plural nil) (interp (train)))
     prep (to ((pn tampa))))

   1: (rule-eval (fire-one
                  ((if% is-a obj-of-prep location)
                   (meaning-of-verb-is ptrans-1) (semantic-role-is to-loc))
                  ((and (if% is-a subj human)
                        (if% is-a obj-of-prep human))
                   (meaning-of-verb-is transport)
                   (thematic-role-is recipient))))
   1: returned ((meaning-of-verb-is ptrans-1) (semantic-role-is to-loc))

 0: returned
    g4946
    (subj ((parse ((pn peter))) (interp (peter))
           (thematic-role (actor) (theme)))
     verb ((main-verb take took) (tense sp) (prim (ptrans-1)))
     obj1 ((parse ((dfart the) (noun train))) (ref (definite))
```

```
          (plural nil) (interp (train)) (thematic-role (instrument)))
     prep ((parse (to ((pn tampa)))) (interp (tampa))
           (thematic-role (to-loc)) (attach-to (verb (strongly)))))
```

Remarks: this time a prepositional phrase is found. The algorithm retrieves the VM to-rules under ‘‘took’’ and tries them. The first such rule fires, so ‘‘rule-eval’’ identifies the meaning of the verb as ‘‘ptrans-1’’ and the thematic role of the prepositional phrase as the ‘‘to-loc.’’ Now that the meaning of the verb is known, the roles of the other syntactic relations can be determined. Thus, the subject of the sentence becomes the ‘‘actor’’ and the ‘‘theme’’ and the object becomes the ‘‘instrument.’’ Next, the parser finds the end of the clause, and calls on the interpreter to execute the end of clause actions. Since nothing in the sentence remains to be interpreted, the interpreter returns the structures unchanged.

Example 2.

```
>>> mary took the man she loved in orlando to tampa.

 0: (interpret g5152 =str obj1 ((dfart the) (noun man)))

    g5152
    (subj ((parse ((pn mary))) (interp (mary)))
     verb ((main-verb take took) (tense sp))
     obj1 ((dfart the) (noun man)))

   1: (rule-eval (fire-one
                  ((if% is-a obj1 bath) (meaning-of-verb-is bathe)
                   (thematic-role-is none))
                  ((if% is-a obj1 trip) (meaning-of-verb-is ptrans-1)
                   (thematic-role-is none))))
   1: returned nil

 0: returned
    g5152
    (subj ((parse ((pn mary))) (interp (mary)))
     verb ((main-verb take took) (tense sp))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man))))
```

Remarks: here, ‘‘interpret’’ is called when the object of the clause is found (the previous calls for ‘‘subj’’ and ‘‘verb’’ are omitted for brevity). The VM object-rules under ‘‘took’’ are unable to determine the meaning of the verb, so control is returned to the main parser algorithm.

```
 0: (interpret g5201 =str subj ((pron she)))

    g5201
    (subj ((pron she)))

 0: returned
    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she))))
```

Remarks: the parser realizes the presence of a relative clause, so it creates a new structure, g5201, to parse its syntactic relations. The interpreter is called to interpret the ‘‘subj’’ of the structure g5201. Although only this structure is shown here for brevity, the interpreter has access to the matrix clause g5152 as well.

```
 0: (interpret g5201 =str verb ((main-verb love loved) (tense sp)))

    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she)))
     verb ((main-verb love loved) (tense sp)))

   1: (rule-eval nil)
   1: returned nil

 0: returned
    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she)))
     verb ((main-verb love loved) (tense sp)))
```

Remarks: the verb is now passed to the interpreter. Since no VM verb-rules are found under ‘‘loved’’, the structures are left unchanged.

```
 0: (interpret g5201 =str obj1
              (((parse ((dfart the) (noun man))) (ref (definite))
                (plural nil) (interp (man)))))
    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she)))
     verb ((main-verb love loved) (tense sp))
     obj1 (((parse ((dfart the) (noun man))) (ref (definite))
           (plural nil) (interp (man)))))

   1: (rule-eval
        (fire-one
           ((if% is-a obj1 human) (meaning-of-verb-is love-r))))
   1: returned ((meaning-of-verb-is love-r))
```

```
 0: returned
    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she))
           (thematic-role (experiencer)))
     verb ((main-verb love loved) (tense sp) (prim (love-r)))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man))
          (rela-referent (@x5255 (q (constant))))
          (thematic-role (theme))))
```

Remarks: the object is now passed to the interpreter.
The VM obj1-rules under ''loved'' fire successfully, identifying the meaning of the verb as ''love-r.'' At the same time, the subject of the clause is identified as the ''experiencer'' role and the object as the ''theme.''

```
 0: (interpret g5201 =str prep (in ((pn orlando))))

    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she))
           (thematic-role (experiencer)))
     verb ((main-verb love loved) (tense sp) (prim (love-r)))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man))
          (rela-referent (@x5255 (q (constant))))
          (thematic-role (theme)))
     prep (in ((pn orlando))))

    g5152
    (subj ((parse ((pn mary))) (interp (mary)))
     verb ((main-verb take took) (tense sp))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man)))
     rela g5201)

   1: (rule-eval nil)
   1: returned nil

 0: returned
    g5201
    (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she))
           (thematic-role (experiencer)))
     verb ((main-verb love loved) (tense sp) (prim (love-r)))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man))
          (rela-referent (@x5255 (q (constant))))
          (thematic-role (theme)))
     prep ((parse (in ((pn orlando)))) (interp (orlando))
```

```
        (attach-to (verb (weakly))) (thematic-role (at-loc))))

  g5152
  (subj ((parse ((pn mary))) (interp (mary))) verb
        ((main-verb take took) (tense sp)) obj1
        ((parse ((dfart the) (noun man))) (ref (definite))
         (plural nil) (interp (man)))
        rela g5201)
```

Remarks: here, the prepositional phrase ``in orlando'' is parsed. The prepositional phrase attachment algorithm is called to determine whether to attach it to the verb ``loved'' in the relative clause, or to the verb ``took'' in the matrix clause (both structures are shown above). Note that the prepositional phrase cannot be attached to the noun phrase ``the man'' since this is the relative clause referent. First, the LTM entry of the verbal concept ``love-r'' is examined to see if it strongly claims the preposition ``in''. Because it does not, the algorithm tries to determine if the verb of the matrix clause (g5152) claims it strongly. No VM in-rules are found under ``took'' so the answer is negative. Next, the algorithm determines if the verb of the current clause (g5201) takes the preposition weakly. The answer in this case is yes, so the preposition is attached weakly to the verb of the relative clause, with meaning ``at-loc'', as indicated by the ''attach-to'' and ``thematic-role'' slots. Note that no attempt is made to weakly attach the preposition to the verb of the matrix clause. Preference is given to the subclause (right association).

```
 0: (interpret g5201 =str prep (to ((pn tampa))))
   g5201
   (subj ((parse ((pron she))) (ref (anaphoric)) (interp (she))
          (thematic-role (experiencer)))
    verb ((main-verb love loved) (tense sp) (prim (love-r)))
    obj1 ((parse ((dfart the) (noun man))) (ref (definite))
         (plural nil) (interp (man))
         (rela-referent (@x5255 (q (constant))))
         (thematic-role (theme)))
    prep ((parse (in ((pn orlando)))) (interp (orlando))
          (attach-to (verb (weakly))) (thematic-role (at-loc)))
    prep (to ((pn tampa))))

   g5152
   (subj ((parse ((pn mary))) (interp (mary)))
    verb ((main-verb take took) (tense sp))
    obj1 ((parse ((dfart the) (noun man))) (ref (definite))
         (plural nil) (interp (man)))
    rela g5201)

  1: (rule-eval
```

```
         (fire-one
            ((if% is-a obj-of-prep location)
             (meaning-of-verb-is ptrans-1) (thematic-role-is to-loc))
            ((and (if% is-a subj human)
                  (if% is-a obj-of-prep human))
             (meaning-of-verb-is transport)
             (thematic-role-is recipient))))
   1: returned ((meaning-of-verb-is ptrans-1) (thematic-role-is to-loc))

 0: returned
    g5152
    (subj ((parse ((pn mary))) (interp (mary (q (constant))))
           (thematic-role (actor) (theme)))
     verb ((main-verb take took) (tense sp) (prim (ptrans-1)))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man (q (?))))
          (thematic-role (accompany)))
     rela g5201
     prep ((parse (to ((pn tampa))))
           (interp (tampa (q (constant))))
           (thematic-role (to-loc))
           (attach-to (verb (strongly)))))

    g5201
    (subj ((parse ((pron she))) (ref (anaphoric))
           (interp (she))
           (thematic-role (experiencer)))
     verb ((main-verb love loved) (tense sp)
           (prim (love-r)))
     obj1 ((parse ((dfart the) (noun man))) (ref (definite))
          (plural nil) (interp (man (q (?))))
          (rela-referent (@x5255 (q (constant))))
          (thematic-role (theme)))
     prep ((parse (in ((pn orlando))))
           (interp (orlando (q (constant))))
           (attach-to (verb (weakly))) (thematic-role (at-loc))))
```

Remarks: next, the prepositional phrase ‘‘to Tampa’’ is found, and the attachment algorithm is activated again. In this case, the verb ‘‘loved’’ does not claim the preposition ‘‘to,’’ but the verb ’’took’’ claims it strongly. This was determined by firing the VM to-rules under ‘‘took,’’ which identified the verbal concept and the thematic roles of the other syntactic relations in the main clause. This is shown in the structures g5201 and g5152, after returning from the call to the interpreter.